



SSSSSSSS	YY	YY	SSSSSSSS	GGGGGGGG	EEEEEEEEE	NN	NN
SSSSSSSS	YY	YY	SSSSSSSS	GGGGGGGG	EEEEEEEEE	NN	NN
SS	YY	YY	SS	GG	EE	NN	NN
SS	YY	YY	SS	GG	EE	NN	NN
SS	YY	YY	SS	GG	EE	NNNN	NN
SS	YY	YY	SS	GG	EE	NNNN	NN
SSSSSS	YY	YY	SSSSSS	GG	EEEEEEE	NN	NN
SSSSSS	YY	YY	SSSSSS	GG	EEEEEEE	NN	NN
SS	YY	YY	SS	GG	EE	NN	NNNN
SS	YY	YY	SS	GG	EE	NN	NNNN
SS	YY	YY	SS	GG	EE	NN	NN
SS	YY	YY	SS	GG	EE	NN	NN
SSSSSSSS	YY	YY	SSSSSSSS	GGGGGG	EEEEEEEEE	NN	NN
SSSSSSSS	YY	YY	SSSSSSSS	GGGGGG	EEEEEEEEE	NN	NN

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

(1)	351	BOOSUSEFILE - Use parameter file
(1)	434	BOOSUSEACT - Use active parameters
(2)	459	BOOSWRTACT - Write parameters to system
(3)	522	BOOSWRTCUR - Write Current Parameters
(4)	558	BOOSSENDOPER - Output facility error message to operator
(4)	602	BOOSCONFIGALL - Auto-configure all adapters
(4)	797	AUTOLOG - AUTO ALL /LOG formating
(4)	844	SGNSGET_DEVICE - Locate device database
(4)	944	Reset routines BOOSRESETLIST and BOOSCONRESET and BOOSMSCP_RESET
(4)	1030	BOOSCONADP - Set connect adapter number
(4)	1154	BOOSCONNECT - Connnect specified device and load driver
(4)	1337	BOOSLOAD - Load a driver or misc code if not already loaded
(4)	1346	BOOSRELOAD - Reload a specified driver
(4)	1431	BOOSGIVEHELP - Print Help information

0000 1 .IF NDF,CONFIGSW  
0000 2 .TITLE SYSGEN - SYSGEN UTILITY AND PARAMETER FILE EDITOR  
0000 3 .IFF  
0000 4 .TITLE CONFIGUTL - SYSGEN UTILITIES FOR CONFIGURE PROCESS  
0000 5 .ENDC  
0000 6 .IDENT 'V04-002'  
0000 7 :  
0000 8 \*\*\*\*\*  
0000 9 \*  
0000 10 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 11 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 12 \* ALL RIGHTS RESERVED.  
0000 13 \*  
0000 14 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 15 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 16 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 17 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 18 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 19 \* TRANSFERRED.  
0000 20 \*  
0000 21 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 22 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 23 \* CORPORATION.  
0000 24 \*  
0000 25 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 26 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 27 \*  
0000 28 \*  
0000 29 \*\*\*\*\*  
0000 30 \*  
0000 31 \*\*  
0000 32 : Facility: System generation and initialization  
0000 33 :  
0000 34 : Abstract: SYSGEN is the main routine to provide all SYSBOOT parameter  
0000 35 : alteration commands in an online environment.  
0000 36 :  
0000 37 : Environment:  
0000 38 :  
0000 39 : Author: RICHARD I. HUSTVEDT, Creation date: 4-MAY-1978  
0000 40 :  
0000 41 : MODIFIED BY:  
0000 42 :  
0000 43 :  
0000 44 : V04-002 WHM0011 Bill Matthews 14-Sep-1984  
0000 45 : Changed the defaults for the MSCP command.  
0000 46 :  
0000 47 : V04-C01 WHM0010 Bill Matthews 04-Sep-1984  
0000 48 : Changed IO PRIORITY default for the MSCP command and  
0000 49 : disallow loading of the MSCP server multiple times.  
0000 50 :  
0000 51 : V03-023 WHM0009 Bill Matthews 23-Jul-1984  
0000 52 : Changed defaults for the MSCP command.  
0000 53 :  
0000 54 : V03-022 WHM0008 Bill Matthews 20-Apr-1984  
0000 55 : Removed WRITE CURRENT code that wrote the SYSGEN parameters  
0000 56 : to SYS.EXE.  
0000 57 :  
0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 :  
0000 65 :  
0000 66 :  
0000 67 :  
0000 68 :  
0000 69 :  
0000 70 :  
0000 71 :  
0000 72 :  
0000 73 :  
0000 74 :  
0000 75 :  
0000 76 :  
0000 77 :  
0000 78 :  
0000 79 :  
0000 80 :  
0000 81 :  
0000 82 :  
0000 83 :  
0000 84 :  
0000 85 :  
0000 86 :  
0000 87 :  
0000 88 :  
0000 89 :  
0000 90 :  
0000 91 :  
0000 92 :  
0000 93 :  
0000 94 :  
0000 95 :  
0000 96 :  
0000 97 :  
0000 98 :  
0000 99 :  
0000 100 :  
0000 101 :  
0000 102 :  
0000 103 :  
0000 104 :  
0000 105 :  
0000 106 :  
0000 107 :  
0000 108 :  
0000 109 :  
0000 110 :  
0000 111 :  
0000 112 :  
0000 113 :  
0000 114 :  
0000 115 :  
0000 116 :  
0000 117 :  
0000 118 :  
0000 119 :  
0000 120 :  
0000 121 :  
0000 122 :  
0000 123 :  
0000 124 :  
0000 125 :  
0000 126 :  
0000 127 :  
0000 128 :  
0000 129 :  
0000 130 :  
0000 131 :  
0000 132 :  
0000 133 :  
0000 134 :  
0000 135 :  
0000 136 :  
0000 137 :  
0000 138 :  
0000 139 :  
0000 140 :  
0000 141 :  
0000 142 :  
0000 143 :  
0000 144 :  
0000 145 :  
0000 146 :  
0000 147 :  
0000 148 :  
0000 149 :  
0000 150 :  
0000 151 :  
0000 152 :  
0000 153 :  
0000 154 :  
0000 155 :  
0000 156 :  
0000 157 :  
0000 158 :  
0000 159 :  
0000 160 :  
0000 161 :  
0000 162 :  
0000 163 :  
0000 164 :  
0000 165 :  
0000 166 :  
0000 167 :  
0000 168 :  
0000 169 :  
0000 170 :  
0000 171 :  
0000 172 :  
0000 173 :  
0000 174 :  
0000 175 :  
0000 176 :  
0000 177 :  
0000 178 :  
0000 179 :  
0000 180 :  
0000 181 :  
0000 182 :  
0000 183 :  
0000 184 :  
0000 185 :  
0000 186 :  
0000 187 :  
0000 188 :  
0000 189 :  
0000 190 :  
0000 191 :  
0000 192 :  
0000 193 :  
0000 194 :  
0000 195 :  
0000 196 :  
0000 197 :  
0000 198 :  
0000 199 :  
0000 200 :  
0000 201 :  
0000 202 :  
0000 203 :  
0000 204 :  
0000 205 :  
0000 206 :  
0000 207 :  
0000 208 :  
0000 209 :  
0000 210 :  
0000 211 :  
0000 212 :  
0000 213 :  
0000 214 :  
0000 215 :  
0000 216 :  
0000 217 :  
0000 218 :  
0000 219 :  
0000 220 :  
0000 221 :  
0000 222 :  
0000 223 :  
0000 224 :  
0000 225 :  
0000 226 :  
0000 227 :  
0000 228 :  
0000 229 :  
0000 230 :  
0000 231 :  
0000 232 :  
0000 233 :  
0000 234 :  
0000 235 :  
0000 236 :  
0000 237 :  
0000 238 :  
0000 239 :  
0000 240 :  
0000 241 :  
0000 242 :  
0000 243 :  
0000 244 :  
0000 245 :  
0000 246 :  
0000 247 :  
0000 248 :  
0000 249 :  
0000 250 :  
0000 251 :  
0000 252 :  
0000 253 :  
0000 254 :  
0000 255 :  
0000 256 :  
0000 257 :  
0000 258 :  
0000 259 :  
0000 260 :  
0000 261 :  
0000 262 :  
0000 263 :  
0000 264 :  
0000 265 :  
0000 266 :  
0000 267 :  
0000 268 :  
0000 269 :  
0000 270 :  
0000 271 :  
0000 272 :  
0000 273 :  
0000 274 :  
0000 275 :  
0000 276 :  
0000 277 :  
0000 278 :  
0000 279 :  
0000 280 :  
0000 281 :  
0000 282 :  
0000 283 :  
0000 284 :  
0000 285 :  
0000 286 :  
0000 287 :  
0000 288 :  
0000 289 :  
0000 290 :  
0000 291 :  
0000 292 :  
0000 293 :  
0000 294 :  
0000 295 :  
0000 296 :  
0000 297 :  
0000 298 :  
0000 299 :  
0000 300 :  
0000 301 :  
0000 302 :  
0000 303 :  
0000 304 :  
0000 305 :  
0000 306 :  
0000 307 :  
0000 308 :  
0000 309 :  
0000 310 :  
0000 311 :  
0000 312 :  
0000 313 :  
0000 314 :  
0000 315 :  
0000 316 :  
0000 317 :  
0000 318 :  
0000 319 :  
0000 320 :  
0000 321 :  
0000 322 :  
0000 323 :  
0000 324 :  
0000 325 :  
0000 326 :  
0000 327 :  
0000 328 :  
0000 329 :  
0000 330 :  
0000 331 :  
0000 332 :  
0000 333 :  
0000 334 :  
0000 335 :  
0000 336 :  
0000 337 :  
0000 338 :  
0000 339 :  
0000 340 :  
0000 341 :  
0000 342 :  
0000 343 :  
0000 344 :  
0000 345 :  
0000 346 :  
0000 347 :  
0000 348 :  
0000 349 :  
0000 350 :  
0000 351 :  
0000 352 :  
0000 353 :  
0000 354 :  
0000 355 :  
0000 356 :  
0000 357 :  
0000 358 :  
0000 359 :  
0000 360 :  
0000 361 :  
0000 362 :  
0000 363 :  
0000 364 :  
0000 365 :  
0000 366 :  
0000 367 :  
0000 368 :  
0000 369 :  
0000 370 :  
0000 371 :  
0000 372 :  
0000 373 :  
0000 374 :  
0000 375 :  
0000 376 :  
0000 377 :  
0000 378 :  
0000 379 :  
0000 380 :  
0000 381 :  
0000 382 :  
0000 383 :  
0000 384 :  
0000 385 :  
0000 386 :  
0000 387 :  
0000 388 :  
0000 389 :  
0000 390 :  
0000 391 :  
0000 392 :  
0000 393 :  
0000 394 :  
0000 395 :  
0000 396 :  
0000 397 :  
0000 398 :  
0000 399 :  
0000 400 :  
0000 401 :  
0000 402 :  
0000 403 :  
0000 404 :  
0000 405 :  
0000 406 :  
0000 407 :  
0000 408 :  
0000 409 :  
0000 410 :  
0000 411 :  
0000 412 :  
0000 413 :  
0000 414 :  
0000 415 :  
0000 416 :  
0000 417 :  
0000 418 :  
0000 419 :  
0000 420 :  
0000 421 :  
0000 422 :  
0000 423 :  
0000 424 :  
0000 425 :  
0000 426 :  
0000 427 :  
0000 428 :  
0000 429 :  
0000 430 :  
0000 431 :  
0000 432 :  
0000 433 :  
0000 434 :  
0000 435 :  
0000 436 :  
0000 437 :  
0000 438 :  
0000 439 :  
0000 440 :  
0000 441 :  
0000 442 :  
0000 443 :  
0000 444 :  
0000 445 :  
0000 446 :  
0000 447 :  
0000 448 :  
0000 449 :  
0000 450 :  
0000 451 :  
0000 452 :  
0000 453 :  
0000 454 :  
0000 455 :  
0000 456 :  
0000 457 :  
0000 458 :  
0000 459 :  
0000 460 :  
0000 461 :  
0000 462 :  
0000 463 :  
0000 464 :  
0000 465 :  
0000 466 :  
0000 467 :  
0000 468 :  
0000 469 :  
0000 470 :  
0000 471 :  
0000 472 :  
0000 473 :  
0000 474 :  
0000 475 :  
0000 476 :  
0000 477 :  
0000 478 :  
0000 479 :  
0000 480 :  
0000 481 :  
0000 482 :  
0000 483 :  
0000 484 :  
0000 485 :  
0000 486 :  
0000 487 :  
0000 488 :  
0000 489 :  
0000 490 :  
0000 491 :  
0000 492 :  
0000 493 :  
0000 494 :  
0000 495 :  
0000 496 :  
0000 497 :  
0000 498 :  
0000 499 :  
0000 500 :  
0000 501 :  
0000 502 :  
0000 503 :  
0000 504 :  
0000 505 :  
0000 506 :  
0000 507 :  
0000 508 :  
0000 509 :  
0000 510 :  
0000 511 :  
0000 512 :  
0000 513 :  
0000 514 :  
0000 515 :  
0000 516 :  
0000 517 :  
0000 518 :  
0000 519 :  
0000 520 :  
0000 521 :  
0000 522 :  
0000 523 :  
0000 524 :  
0000 525 :  
0000 526 :  
0000 527 :  
0000 528 :  
0000 529 :  
0000 530 :  
0000 531 :  
0000 532 :  
0000 533 :  
0000 534 :  
0000 535 :  
0000 536 :  
0000 537 :  
0000 538 :  
0000 539 :  
0000 540 :  
0000 541 :  
0000 542 :  
0000 543 :  
0000 544 :  
0000 545 :  
0000 546 :  
0000 547 :  
0000 548 :  
0000 549 :  
0000 550 :  
0000 551 :  
0000 552 :  
0000 553 :  
0000 554 :  
0000 555 :  
0000 556 :  
0000 557 :  
0000 558 :  
0000 559 :  
0000 560 :  
0000 561 :  
0000 562 :  
0000 563 :  
0000 564 :  
0000 565 :  
0000 566 :  
0000 567 :  
0000 568 :  
0000 569 :  
0000 570 :  
0000 571 :  
0000 572 :  
0000 573 :  
0000 574 :  
0000 575 :  
0000 576 :  
0000 577 :  
0000 578 :  
0000 579 :  
0000 580 :  
0000 581 :  
0000 582 :  
0000 583 :  
0000 584 :  
0000 585 :  
0000 586 :  
0000 587 :  
0000 588 :  
0000 589 :  
0000 590 :  
0000 591 :  
0000 592 :  
0000 593 :  
0000 594 :  
0000 595 :  
0000 596 :  
0000 597 :  
0000 598 :  
0000 599 :  
0000 600 :  
0000 601 :  
0000 602 :  
0000 603 :

0000	58	V03-021	WHM0007	Bill Matthews	04-Apr-1984
0000	59			Added support to write current to write to a seperate	
0000	60			default system parameter file.	
0000	61			Added support to use file to accept long ascii sysgen parameters	
0000	62				
0000	63	V03-020	WHM0006	Bill Matthews	14-Mar-1984
0000	64			Modify SGNSGET_DEVICE to take out the I/O database MUTEX and	
0000	65			raise IPL before calling IOC\$SEARCHALL.	
0000	66				
0000	67	V03-019	WHM0005	Bill Matthews	13-Mar-1984
0000	68			Move definition of BOOSGL_LOAD_ARGS from SYSBOOCMD to	
0000	69			this module.	
0000	70				
0000	71	V03-018	ACG0399	Andrew C. Goldstein	10-Mar-1984 0:36
0000	72			Change check for SSS NODEVAVL to SSS_NOSUCHDEV due to	
0000	73			rewrite of IOC\$SEARCHADEV.	
0000	74				
0000	75	V03-016	WHM0004	Bill Matthews	23-Feb-1984
0000	76			Added support for loading and starting the MSCP server.	
0000	77				
0000	78	V03-015	WHM0003	Bill Matthews	04-Feb-1984
0000	79			Added support for ACF\$B_COMBO_VECTOR_OFFSET to clean up support	
0000	80			of combo style devices.	
0000	81				
0000	82	V03-014	TMK0001	Todd M. Katz	31-Jan-1984
0000	83			Change a BSBW to a JSB.	
0000	84				
0000	85	V03-013	WHM0002	Bill Matthews	13-Dec-1983
0000	86			Fixed several calls to SGNSGET_DEVICE to pass the unit number	
0000	87			to be connected not the maximum units.	
0000	88			Added support for the new CONNECT command qualifiers	
0000	89			/CSR_OFFSET and /VECTOR_OFFSET.	
0000	90				
0000	91	V03-012	JLV0312	Jake VanNoy	26-Oct-1983
0000	92			Fix bug for microVAX that allows nexus 0 in CONNECT.	
0000	93				
0000	94	V03-011	WHM0001	Bill Matthews	09-Dec-1983
0000	95			Changed some bsbw's to jsb's	
0000	96				
0000	97	V03-010	WMC0003	Wayne Cardoza	09-Aug-1983
0000	98			Fix loadable code error handling.	
0000	99			USEACTIVE should be in configutl.	
0000	100				
0000	101	V03-009	WMC0002	Wayne Cardoza	29-Jul-1983
0000	102			More features for code loading.	
0000	103				
0000	104	V03-008	WMC0001	Wayne Cardoza	27-Jul-1983
0000	105			Support general code loading.	
0000	106				
0000	107	V03-007	MSH0006	Maryann Hinden	24-Jun-1983
0000	108			Use \$BOOCMDDEF instead of \$BOODEF.	
0000	109				
0000	110	V03-006	MSH0005	Maryann Hinden	04-May-1983
0000	111			Changes to support CONFIGURE process.	
0000	112				
0000	113	V03-005	MSH0004	Maryann Hinden	13-May-1983
0000	114			Change some BSBW PUTERROR instructions to JSB instead.	

0000	115				
0000	116	V03-004 MSH0003	Maryann Hinden	31-Jan-1983	
0000	117	Add support for cluster device names.			
0000	118				
0000	119	V03-003 TCM0001	Trudy C. Matthews	8-Nov-1982	
0000	120	Use new ADPSL_AVECTOR field in calculation of ACF\$W_AVECTOR,			
0000	121	instead of calculating it from the adapter's TR number.			
0000	122				
0000	123	V03-002 MSH0002	Maryann Hinden	22-Oct-1982	
0000	124	Fix broken BSBW.			
0000	125				
0000	126	V03-001 MSH0001	Maryann Hinden	30-Sep-1982	
0000	127	Check for DDBSL_UCB = 0.			
0000	128	--			
0000	129				
0000	130				
0000	131	Include files:			
0000	132				
0000	133	SACFDEF	:	Define autoconfiguration block	
0000	134	SADPDEF	:	Define adapter control block	
0000	135	SBOOCMDDEF	:	Define SYSGEN command options	
0000	136	SCLIDEF	:	Define CLI codes and values	
0000	137	SCRBDEF	:	Define CRB offsets	
0000	138	SDBBDEF	:	Define DDB offsets	
0000	139	SDYNDEF	:	Block types	
0000	140	SHLPDEF	:	Define HELP symbols	
0000	141	\$IDBDEF	:	Define IDB offsets	
0000	142	\$IHDEF	:	Image header offsets	
0000	143	\$IPLDEF	:	Define IPLs	
0000	144	\$JPIDEF	:	\$GETJPI definitions	
0000	145	\$LBRDEF	:	Librarian symbols	
0000	146	\$OPCDEF	:	Operator message definitions	
0000	147	\$PRDEF	:	Define processor registers	
0000	148	\$PRMDEF	:	Parameter descriptor definitions	
0000	149	\$SBDEF	:	SCS system block definitions	
0000	150	\$SHRDEF	:	Error codes	
0000	151	\$SSLVDEF	:	Loadable code header	
0000	152	\$SSDEF	:	Define system status values	
0000	153	\$SYSGMSGDEF	:	Sysgen messages	
0000	154	\$TPADEF	:	TPARSE definitions	
0000	155	\$UCBDEF	:	Define UCB offsets	
0000	156	\$VECDEF	:	Define VEC offsets	
0000	157				
0000	158				
0000	159	Equated Symbols:			
0000	160				
0000000D	0000	161	CR=13	:	Character value for carriage return
0000000C	0000	162	FF=12	:	Character value for form feed
0000000A	0000	163	LF=10	:	Character value for line feed
00001000	0000	164	UBA_IOPAGE=8*512	:	Offset from UBA configuration register
	0000	165		:	to base of I/O page
	0000	166			
	0000	167	Own Storage		
	0000	168			
00000000	0000	169	.PSECT \$\$\$\$0000,NOEXE,NOWRT	:	PSECT to mark lower address
00000000	0000	170	BOOSLOLIM::	:	Marker definition
00000000	0000	171	.PSECT ____ZZZ,WRT,PAGE	:	PSECT to mark upper address limit

0000	0000	172	BOOSHILIM::	
00000000	0000	173	.PSECT NONPAGED_DATA	rd,wrt,noexe,quad
0000	0000	174		
00000200	0000	175	BOOSAB_PATCH::	: Non-paged Patch area
0200	0200	176	.BLKB 512	: One page
00002200	0200	177	BOOSAB_PRMBUF::	: Parameter buffer
2200	2200	178	.BLKB 512*16	: A generous buffer
00002400	2200	179	BOOSAB_LOADBUF::	: Buffer for code loader
2400	180		.BLKB 512	
00000000	2400	181	ACFSGL_DDB::	
00000000	2400	182	.LONG 0	
2404	183		ACFSGL_UCB::	
00000000	2404	184	.long 0	
2408	185		ACFSGL_IDB::	
00000000	2408	186	.long 0	
240C	187		ACFSGL_CRB::	
00000000	240C	188	.long 0	
2410	189		ACFSGL_LASTDDB::	
00000000	2410	190	.long 0	
2414	191		ACFSGL_DPT::	
00000000	2414	192	.long 0	
2418	193		ACFSGL_SB::	
00000000	2418	194	.LONG 0	
241C	195		BOOSGL_COMBO_VECTOR_OFFSET::	: Offset to vector from start of combo
00000000	241C	196	.LONG 0	: device's vectors
2420	197		BOOSGL_COMBO_CSR_OFFSET::	: Offset to CSR from start of combo
00000000	2420	198	.LONG 0	: device's CSR
2424	199		BOOSGL_CONADP::	: Adapter TR number
FFFFFFFFFF	2424	200	.LONG -2	: Null value
FFFFFFFFFF	2428	201	BOOSGL_CONCREG::	: Control register
FFFFFFFFFF	2428	202	.LONG -1	: Null value
FFFFFFFFFF	242C	203	BOOSGL_CONCUNIT::	: Controller unit
FFFFFFFFFF	242C	204	.LONG -1	: Null value
2430	205		BOOSGL_CONNUMU::	: Number of Units to configure
00000001	2430	206	.LONG 1	: Default value is 1 unit
FFFFFFFFFF	2434	207	BOOSGL_CONVECT::	: Vector offset
FFFFFFFFFF	2434	208	.LONG -1	: Null value
FFFFFFFFFF	2438	209	BOOSGL_CONNUMV::	: Number of vectors
FFFFFFFFFF	2438	210	.LONG -1	: Null value
FFFFFFFFFF	243C	211	BOOSGL_CONAUNIT::	: Adapter unit
FFFFFFFFFF	243C	212	.LONG -1	: Null value
FFFFFFFFFF	2440	213	BOOSGL_CONDEV::	: Device name string address
FFFFFFFFFF	2440	214	.LONG -1	: Null value
FFFFFFFFFF	2444	215	BOOSGL_CONDRV::	: Driver name string address
FFFFFFFFFF	2444	216	.LONG -1	: Null value
00000000	2448	217	BOOSGL_CONUNITS::	: Maximum units
00000000	2448	218	.LONG 0	
00000000	244C	219	BOOSGQ_CONSYSID::	: System ID
00000000	244C	220	.LONG 0	: quadword
00000000	2450	221	.LONG 0	
00000000	2454	222	BOOSGL_CONCRB::	: CRB address
00000000	2454	223	.LONG 0	
00000000	2458	224	BOOSGL_CONFLAGS::	: Flags
00000000	2458	225	.LONG 0	
00000000	245C	226	BOOSGL_NEXTSTR::	: Next string location
00000000	245C	227	.LONG 0	
2460	228		BOOSGL_SELECT::	: Address of select list

00 20 20 3E 4E 45 47 53 59 53 0A 0D	2460 229 .LONG 0	: CLI call back block
0000246C	2464 230 BOOSAL_CLIBLK::	: Get command call back block
	2464 231 SCLIREQDESC -	
	2464 232 RQTYPE=CLISK_GETCMD	
	000024B4 2480 233 BOOSGQ_CMDESC==BOOSAL_CLIBLK+CLISW_ROSIZE	
	2480 234 BOOSGT_PROMPT::	; Command descriptor address
	2480 235 .ASCIZ <CR><LF>%SYSGEN> %	Prompt string
	000024B4 248C 236 BOOSAL_ACF::	
	248C 237 .BLKB ACFSC_LENGTH	: Auto-configuration block
	2484 238 BOOSGQ_LIMITS::	: Allocate space for it
	2484 239 .LONG BOOSLOLIM	: High and low address limits for lockdown
	2484 240 .LONG BOOSHILIM-1	: Lower address bound
00000000 00000000	24BC 241 BOOSGQ_RETADR::	: Return address receiver
	24BC 242 .LONG 0,0	
00000000 00000000	24C4 243 BOOSGL_RETSAVE::	: Saved co-routine return address
	24C4 244 .LONG 0	
000024D0'00000006' 4E 45 47 53 59 53 00000006	24C8 245 FACNAMED::	: Facility name descriptor
	24C8 246 .LONG FACNAMSZ,FACNAME	
	24D0 247 FACNAME:.ASCII /SYSGEN/	
	24D6 248 FACNAMSZ=.FACNAME	
41 53 43 00' 03	24D6 249 CONNAME:	
	24D6 250 .ASCIC /CSA/	: device name
41 50 4F 00' 03	24DA 251 BOOSGT_OPNAME::	: Console terminal device name
	24DA 252 .ASCIC /OPA/	
52 45 56 49 52 44 56 43 00' 08	24DE 253 BOOSGT_CVNAME::	: Name of RL02 driver
	24DE 254 .ASCIC /CVDRIVER/	
52 45 56 49 52 44 58 44 00' 08	24E7 255 BOOSGT_DXNAME::	: Name of floppy driver
	24E7 256 .ASCIC /DXDRIVER/	
52 45 56 49 52 44 44 44 00' 08	24F0 257 BOOSGT_DDNAME::	: Name of TU58 driver
	24F0 258 .ASCIC /DDDRIVER/	
00000000 00	24F9 259 BOOSGL_FILEADDR::	: File spec address
	24F9 260 .LONG 0	
	24FD 261 BOOSGB_FILELEN::	: File spec length
	24FD 262 .BYTE 0	
74 6E 65 72 72 75 43 00' 07	24FE 263	
65 76 69 74 63 41 00' 06	2502 265 BOOSGL_PARINUSE:: .LONG 0	
74 6C 75 61 66 65 44 00' 07	2502 266 BOOSGT_CURRENT:: .ASCIC /Current/	
00002559	250A 267 BOOSGT_ACTIVE:: .ASCIC /Active/	
45 48 24 53 59 53 00002561'010E0000' 4C 48 2E 4E 45 47 53 59 53 3A 50 4C 42	2511 268 BOOSGT_DEFAULT:: .ASCIC /Default/	
00000001 00002580'010E0000'	2519 269 BOOSGT_FILE:: .BLKB 64	
	2559 270 HELPFILE: .ASCID /SYSSHELP:SYSGEN.HLB/	: Help library file name
	2559 271 .ASCID //	
	2567 273 HELPFLAG: .long hlp\$M_prompt	
	2573 274 HELP_DESC: .ascid //	: Filled in as pointer
	2580 275	

```

00000000 2580 276 VALID_PARFILE: : Valid parameter file flag
00000000 2580 277 .LONG 0
00000000 2584 278 SAVE_DOT: : Save dot through USE filespec
00000000 2584 279 .LONG 0
00000000 2588 280 FULL_NAME_PTR:: : Full device name
00000000 2588 281 .LONG 0
00000000 258C 282
00000000 258C 283 : MSCP initialization routine default argument list
00000000 258C 284
00000008 258C 285 MSCP_ARG_LIST: : Number of arguments
00000001 2590 286 .LONG 8 : Function code(load and start server)
00008000 2594 287 .LONG 1 : Default buffer size
00000004 2598 288 .LONG 32768 : Default number of receive credits for each host
0000000F 259C 289 .LONG 4 : Default number of hosts supported
00000014 25A0 290 .LONG 15 : Default time out
00000004 25A4 291 .LONG 20 : Default priority
00001000 25AB 292 .LONG 4 : Default for minimum qualifier
00004000 25AC 293 .LONG 4096 : Default for maximum qualifier
000025BC 25B0 294 .LONG 16384 : Space for new args
00000030 25B0 295 .BLKL 3
00000000 25B0 296 MSCP_ARG_LIST_SIZE = .-MSCP_ARG_LIST
000025EC 25B0 297
000025EC 25B0 298 BOOSGL_LOAD_ARGS:: : Argument list block loadable code init
000025EC 25B0 299 .BLRB MSCP_ARG_LIST_SIZE : routine
000025EC 25EC 300
000025EC 25EC 301
50 43 53 4D 00 04 25EC 302 MSCP_NAME: .ASCIC /MSCP/ ; MSCP server name
000025EC 25EC 303
000025EC 25F1 304 : AUTO ALL /LOG storage
000025EC 25F1 305
000025EC 25F1 306 CTRSTR_AUTOLOG: .ascid / !AC!UW/
000025EC 25FF 307 CTRSTR_AUTOLOG_UNIT: .ascid /,!UW/
000025EC 2600 308 Outlen_unit: .long 0
000025EC 260C 309 Outlen: .long 0
000025EC 2610 310 Boo$gt_save_devname: .blk 20
000025EC 2614 311 outbuf: .ascid //.
000025EC 2628 312 outbuf_str: .blk 100
000025EC 2630 313
000025EC 2694 314 : Send operator message data
000025EC 2694 315
000025EC 2694 316 OPERGETJPI: : $GETJPI item list
0004 2694 317 .WORD 4 : Buffer length
0319 2696 318 .WORD JPI$ PID : Process ID code
00002680 2698 319 .ADDRESS OPERMSGPID : Buffer address
00000000 269C 320 .LONG 0 : Don't return length
00000000 26A0 321 .LONG 0 : List terminator
00002680 26A4 322
0003 26A4 323 OPERMSGVEC: : $PUTMSG message vector
000F 26A6 324 .WORD 3 : Argument count
00000000 26A8 325 .WORD ^B1111 : Default message flags
00000000 26A8 326 OPERMSGID: : Message ID
0001 26AC 327 .LONG 0
0000 26AC 328 OPERMSGFAO: : FAO argument count
0000 26AE 329 .WORD 1
0000 26AE 330 .WORD 0 : No new message flags

```

00000000	2680	331	OPERMSGID:		;	PID of this process		
	2680	332		.LONG	0			
000026B8	2684	333	OPERMSGNAM:		;	File specification		
	2684	334		.ADDRESS	OPERNAMDESC			
00000000 00000000	2688	335	OPERNAMDESC:					
	2688	336		.LONG	0,0			
	26C0	337	OPERNAMDESC:					
	26C0	338		.LONG	0,0			
00000000	26C0	339	OPERMSG:		;	Message descriptor		
	26C0	340		.LONG	0			
000026C8	26C4	341		.ADDRESS	OPERMMSGBUF			
	26C8	342	OPERMMSGBUF:		;	Message buffer		
00000103	26C8	343		.LONG	OPC\$_RQ_RQST!<OPCSM_NM_CENTRL88>	;	Message type and target	
00000000	26CC	344		.LONG	0	;	No reply message	
000027D0	26D0	345	OPERMMSGTXT:				;	Message text
	26D0	346		.BLKB	256			
	27D0	347						
	27D0	348						
	27D0	349		.IF	NDF,CONFIGSW	;	SYSGEN-specific code	

27D0 351 .SBTTL BOOSUSEFILE - Use parameter file  
 27D0 352 ++  
 27D0 353 Functional description:  
 27D0 354 BOOSUSEFILE reads the specified file in response to the USE  
 27D0 355 command and merges all of the values specified in that file into  
 27D0 356 the working copy of the parameter values. This is accomplished  
 27D0 357 by looking up each value specified and merging the associated  
 27D0 358 value.  
 27D0 359  
 27D0 360 Calling sequence:  
 27D0 361 CALLG arglist,BOOSUSEFILE  
 27D0 362  
 27D0 363 Input Parameters:  
 27D0 364 TPASL\_TOKENCNT(AP) - Length of file name string  
 27D0 365 TPASL\_TOKENPTR(AP) - Address to file name string  
 27D0 366 Output Parameters:  
 27D0 367 R0 - Completion status code  
 27D0 368  
 27D0 369 --  
 27D0 370  
 00000000 371 .PSECT PAGED\_CODE rd,nowrt,exe,long  
 0000 372  
 03FC 0000 373 .Entry BOOSUSEFILE, ^M<R2,R3,R4,R5,R6,R7,R8,R9> ; Entry mask  
 0000 374  
 0002 375  
 0002 376 BBSS #EXESV WRITESYSPARAMS,- : Use a file => write current needed  
 00 00000000'8F 0002 377 G^EXESGL\_DYNAMIC\_FLAGS,1\$;  
 00000000'GF 0008 378 1\$:  
 00002584'EF 00000000'EF 00 379 MOVL BOOSGL\_DOT,L^SAVE\_DOT : Save dot  
 57 10 AC 9E 0019 380 MOVAB TPASL\_TOKENCNT(AP),R7 : Set address of file name descriptor  
 FFE0. 30 001D 381 BSBW BOOS\$FILEOPEN : Open specified file  
 04 50 E8 0020 382 BLBS R0,20\$ : Continue if success  
 50 01 3C 0023 383 MOVZWL #1,R0 : Force success  
 04 0026 384 RET  
 56 00000200'EF 9E 0027 385 10\$: MOVAB BOOS\$AB\_PRMBUF,R6 : Set address of parameter buffer  
 59 10 D0 002E 386 MOVL #16,R9 : Set size of buffer  
 FFFC. 30 0031 387 BSBW BOOS\$READFILE : Read file content into parameter buffer  
 EC 50 E9 0034 388 BLBC R0,10\$ : Exit if error  
 58 00000200'EF 9E 0037 389 MOVAB BOOS\$AB\_PRMBUF,R8 : Init pointer to parameter buffer  
 00000000'EF 68 20 28 003E 390 MOVC3 #32,(R8),EXESGT\_STARTUP : Set startup command file name  
 58 20 C0 0046 391 ADDL #32,R8 : and advance buffer pointer  
 00002580'EF D4 0049 392 CLRL VALID\_PAR\_FILE : Initialize valid parameter file flag  
 68 D5 004F 393 30\$: TSTL (R8) : Check for end of list  
 56 13 0051 394 BEQL DONE : Branch if yes  
 10 AC 68 9A 0053 395 MOVZBL (R8),TPASL\_TOKENCNT(AP) : Set token count for search  
 14 AC 01 A8 9E 0057 396 MOVAB 1(R8),TPASL\_TOKENPTR(AP) : And address of string  
 58 10 C0 005C 397 ADDL #16,R8 : Advance to value  
 1C AC 88 D0 005F 398 MOVL (R8)+,TPASL\_NUMBER(AP) : Set number  
 00000000'EF 6C FA 0063 399 CALLG (AP),L^BOOSSEARCH : Search for parameter  
 E2 50 E9 006A 400 BLBC R0,30\$ : Next parameter if not found  
 00002580'EF 01 D0 006D 401 MOVL #1,VALID\_PAR\_FILE : Indicate valid parameter file  
 54 20 AC D0 0074 402 MOVL TPASL\_PARAM(AP),R4 : Get a pointer to the parameter descriptor  
 22 10 A4 10 E1 0078 403 BBC #PRMSD ASCII,PRMSL\_FLAGS(R4),40\$; Branch if not an ascii parameter  
 14 AC 78 DE 007D 404 MOVAL -(R8),TPASL\_TOKENPTR(AP) : Get a pointer to the parameter value  
 50 14 A4 9A 0081 405 MOVZBL PRMSB\_SIZE(R4),R0 : Get parameter size in bits  
 50 50 FD 8F 78 0085 406 ASHL #3,R0,R0 : Set parameter size  
 10 AC 50 9A 008A 407 MOVZBL R0,TPASL\_TOKENCNT(AP) :

50 03	CO 008E	408	ADDL2	#3,R0	;	Round size up to the next longword	
50 03	CA 0091	409	BICL2	#3,R0	;	Advance past value	
58 50	CO 0094	410	ADDL2	R0,R8	;	Set the value of the parameter	
0000'CF	6C FA	0097	411	CALLG	(AP),W"BOOSSETASCII	;	Continue with the next parameter
00000000'EF	FFB0 31	009C	412	BRW	30\$	;	Set value of parameter
FFA6 6C	FA 009F	413	40\$:	CALLG	(AP),L"BOOSSETVALUE	;	Continue with next parameter
FF54' 31	00A6	414	BRW	30\$	;	Close the file	
09 00002580'EF	E8 00AC	415	DONE:	BSBW	BOOSFILECLOSE	;	If LBS, valid parameter file
50 007C80EA 8F	DO 00B3	416	BLBS	VALID PAR FILE,10\$	;	Set error	
21 11	00BA	417	MOVL	#SYSGE_NOTPARAM,R0	;	Set error	
		418	BRB	20\$	;	Branch	
		419	10\$:				
		420					
		421					
		422					
58 00002519'EF	DE 00BC	423	MOVAL	BOOSGT FILE,R8	;	Set address of String	
000024FE'EF	58 DO	00C3	424	MOVL	R8,BOOSGL_PARINUSE	;	Set address
68 000024FD'EF	9A 00CA	425	MOVZBL	BOOSGB FILELEN,(R8)	;	Set count	
000024F9'FF	68 28	00D1	426	MOVCS	(R8),#BOOSGL_FILEADDR,-	;	Move string
01 A8	00D8	427		1(R8)			
		428					
		429					
00000000'EF	50 01	3C 00DA	430	MOVZWL	#SSS_NORMAL,R0	;	Return success
00002584'EF	DO 00DD	430	20\$:	MOVL	L"SAVE_DOT,BOOSGL_DOT	;	Restore dot
	04 00E8	431		RET		;	
	00E9	432		.ENDC		;	End of SYSGEN-specific code

00E9 434 .SBTTL BOOSUSEACT - Use active parameters  
00E9 435 ++  
00E9 436 Functional description:  
00E9 437 This routine copies the parameter values from the running  
00E9 438 system to the working copy of the parameter values.  
00E9 439 Calling sequence:  
00E9 440  
00E9 441 CALLS #0,BOOSUSEACT  
00E9 442  
00E9 443 Input parameters:  
00E9 444 None  
00E9 445 Output Parameters:  
00E9 446 R0 - Completion status code  
00E9 447 :--  
00E9 448  
003C 00E9 449 .Entry BOOSUSEACT,^M<R2,R3,R4,R5>  
00000000'EF 0000'8F 28 00EB 450  
00000000'EF 0000250A'EF DE 00F9 451  
000024FE'EF 50 01 D0 0104 452  
04 0107 453  
0108 454  
0107 455  
0108 456  
0108 457  
00EB 451  
00EF 452  
00F9 453  
00FF 454  
0104 455  
0107 456  
0108 457  
MOV3 #EXESC,SYSPARSZ,- : Move parameters  
MMGSA,SYSPARAM,EXESA\_SYSPARAM  
MOVAL BOOSGT\_ACTIVE,-  
BOOSGL\_PARINUSE : Set parameter in use  
MOVL #1,R0 : Return success  
RET  
.IF NDF,CONFIGSW : SYSGEN-specific code

```

0108 459 .SBTTL BOOSWRTACT - Write parameters to system
0108 460 ++
0108 461 Functional Description:
0108 462 This routine writes the parameters in the working parameter
0108 463 buffer to the system's parameter area. Only dynamic
0108 464 parameters are copied.
0108 465
0108 466 Calling Sequence:
0108 467 CALLS #0,BOOSWRTACT
0108 468
0108 469 Input Parameters:
0108 470 None
0108 471
0108 472 Output Parameters:
0108 473 R0 - Completion status code
0108 474 :--+
0108 475 00000000
0108 476 .PSECT NONPAGED_CODE rd,nowrt,exe,long
0108 477
0000 0000 0002 .Entry BOOSWRTACT, ^M<>
0000 0002 479
0000 0002 480 SCMKRNL_S B^10$, (AP)
0000 0002 481 BLBC R0,1$ ; Do it in kernel mode
0000 0002 482 JSB BOOSSENDOPER ; If LBC, error
0000 0002 483 .LONG SYSGS_WRITEACT ; Notify operator of WRITE ACTIVE
0000 0002 484 BLBS R0,5$ ; If LBS, success
0000 0002 485 1$: JSB PUTERROR ; Report error
0000 0002 486 MOVL #1,R0 ; Force success
0000 0002 487 5$: RET
0000 0002 488
0000 0002 489 10$: .WORD ^M<R2,R3,R4,R5>
0000 0002 490 MOVAB L^BOOSA_PRMBLK,R5 ; Get base of parameter blocks
0000 0002 491 DSBINT #IPL8_SCHED ; Raise IPL to prevent being unscheduled
0000 0002 492
0000 0002 493
0000 0002 494 ASSUME PRMSL_ADDR EQ 0 ; (Assumes pages are locked in W.S.)
0000 0002 495
0000 0002 496 20$: MOVL PRMSL_ADDR(R5),R3 ; Get address of parameter
0000 0002 497 BEQL 40$ ; Reached the end
0000 0002 498 BBC #PRMSV_DYNAMIC- ; Branch if this is not a
0000 0002 499 PRMSL_FLAGS(R5),30$ ; dynamic parameter
0000 0002 500 MOVZBL PRMSB_POS(R5),R1 ; Get position of parameter
0000 0002 501 EXTZV R1,PRMSB_SIZE(R5), (R3),R2 ; Extract parameter value
0000 0002 502 MOVAB L^EXESA_SYSPARAM,R0 ; Get address of working buffer
0000 0002 503 SUBL R0,R3 ; Get parameter offset
0000 0002 504 INSV R2,R1,PRMSB_SIZE(R5),- ; Store in system
0000 0002 505 L^MMGSA_SYSPARAM(R3)
0000 0002 506
0000 0002 507 30$: ADDL #PRMSL_LENGTH,R5 ; Point to next parameter block
0000 0002 508 BRB 20$ ; Repeat
0000 0002 509
0000 0002 510 ; Copy dynamic flags from default flags to R0
0000 0002 511
0000 0002 512 40$: BICL3 #^C<PRMSL_DYNFLAGS>- ; Clear dynamic flags in real flags
0000 0002 513 MMGSA_SYSPARAM+<EXE$GL_DEFFLAGS-EXESA_SYSPARAM>,R0
0000 0002 514 BICL #PRMSL_DYNFLAGS,- ; EXE$GL_FLAGS
0000 0002 515
55 00000000'EF 003C 0028 488
55 00000000'EF 9E 002A 490
55 00000000'EF 003C 002A 490
55 00000000'EF 9E 0031 491
55 00000000'EF 003C 002A 491
55 00000000'EF 9E 0037 492
55 00000000'EF 003C 002A 492
55 00000000'EF 9E 0037 493
55 00000000'EF 003C 002A 493
55 00000000'EF 9E 0037 494
55 00000000'EF 003C 002A 494
55 00000000'EF 9E 0037 495
55 00000000'EF 003C 002A 495
55 00000000'EF 9E 0037 496
55 00000000'EF 003C 002A 496
55 00000000'EF 9E 0037 497
55 00000000'EF 003C 002A 497
55 00000000'EF 9E 0037 498
55 00000000'EF 003C 002A 498
55 00000000'EF 9E 0037 499
55 00000000'EF 003C 002A 499
55 00000000'EF 9E 0041 500
55 00000000'EF 003C 003E 500
55 00000000'EF 9E 0041 500
55 00000000'EF 003C 003E 500
55 00000000'EF 9E 0045 501
55 00000000'EF 003C 0041 501
55 00000000'EF 9E 0045 501
55 00000000'EF 003C 0041 501
55 00000000'EF 9E 004B 502
55 00000000'EF 003C 0045 502
55 00000000'EF 9E 004B 502
55 00000000'EF 003C 0045 502
55 00000000'EF 9E 0052 503
55 00000000'EF 003C 0052 503
55 00000000'EF 9E 0052 503
55 00000000'EF 003C 0052 503
55 00000000'EF 9E 0055 504
55 00000000'EF 003C 0055 504
55 00000000'EF 9E 0055 504
55 00000000'EF 003C 0055 504
55 00000000'EF 9E 005A 505
55 00000000'EF 003C 005A 505
55 00000000'EF 9E 005A 505
55 00000000'EF 003C 005A 505
55 00000000'EF 9E 005F 506
55 00000000'EF 003C 005F 506
55 00000000'EF 9E 005F 506
55 00000000'EF 003C 005F 506
55 00000000'EF D3 005F 507
55 00000000'EF 11 005F 507
55 00000000'EF D3 0062 508
55 00000000'EF 11 0062 508
55 00000000'EF D3 0064 509
55 00000000'EF 11 0064 509
55 00000000'EF D3 0064 510 ; Copy dynamic flags from default flags to R0
55 00000000'EF 11 0064 511
55 00000000'EF CB 0064 512 40$: BICL3 #^C<PRMSL_DYNFLAGS>- ; Clear dynamic flags in real flags
55 00000000'EF CA 006A 513 MMGSA_SYSPARAM+<EXE$GL_DEFFLAGS-EXESA_SYSPARAM>,R0
55 00000000'EF CA 0070 514 BICL #PRMSL_DYNFLAGS,- ; EXE$GL_FLAGS
55 00000000'EF CA 0076 515

```

00000000'EF	50	C8	007B	516	BISL	R0,EXE\$GL_FLAGS	; Set dynamic flags in real flags
			0082	517			
			0085	518	ENBINT		; Lower IPL
	50	01	0085	519	MOVL	#1,R0	; Set success
		04	0088	520	RET		

0089 522 .SBTTL BOOSWRTCUR - Write Current Parameters  
 0089 523 ++  
 0089 524 Functional Description:  
 0089 525 This routine writes the parameters from the working parameter  
 0089 526 buffer to the system parameter file on disk. They will take effect the  
 0089 527 next time the system is booted.  
 0089 528  
 0089 529 Calling Sequence:  
 0089 530 CALLS #0,BOOSWRTCUR  
 0089 531  
 0089 532 Input parameters:  
 0089 533 None  
 0089 534  
 0089 535 Output Parameters:  
 0089 536 R0 - Completion status code  
 0089 537 :--  
 0089 538  
 00000108 539 .PSECT PAGED\_CODE rd,nowrt,exe,long  
 0108 540  
 03FC 0108 541 .Entry BOOSWRTCUR, ^M<R2,R3,R4,R5,R6,R7,R8,R9>  
 010A 542  
 00 00000000'BF E5 010A 543 BBCC #EXESV WRITESYSPARAMS,- : Don't do WRITE CURRENT again in startup  
 00 00000000'GF 0110 544 G^EXESGL DYNAMIC FLAGS,10\$:  
 50 00000000'EF 9E 0116 545 10\$: MOVAB BOOSGT SYSPARNAME, R0 : Get address of system .PAR file name  
 10 AC 80 9A 011D 546 MOVZBL (R0)+,TPASL TOKENCNT(AP) : Set up for call to BOOSWRTSYSPARFILE  
 14 AC 50 D0 0121 547 MOVL R0,TPASL TOKENPTR(AP)  
 00000000'GF 6C FA 0125 548 CALLG (AP),G^BOOSWRTSYSPARFILE : Call the routine to write out the file  
 0A 50 E9 012C 549 BLBC R0,20\$ : Branch if error  
 000E 30 012F 550 BSBW BOOSSENDOPER : Notify operator of WRITE CURRENT  
 007CA01B 0132 551 .LONG SYSGS\_WRITECUR  
 03 50 E8 0136 552 BLBS R0,30\$ : If LBS, success  
 FEC4' 30 0139 553 20\$: BSBW PUTERROR : Report error  
 50 01 D0 013C 554 30\$: MOVL #1,R0 : Return success  
 04 013F 555 RET  
 0140 556

0140 558 .SBTTL BOOSSENDOPER - Output facility error message to operator  
 0140 559  
 0140 560 : Functional Description:  
 0140 561 : BOOSSENDOPER outputs an error message to the operator.  
 0140 562  
 0140 563 : Calling Sequence:  
 0140 564 BSBW BOOSSENDOPER  
 0140 565 .LONG <msg-id>  
 0140 566  
 0140 567 BOOSSENDOPER:::  
 000026A8'EF 00 BE D0 0140 568 MOVL 2(SP), OPERMSGID : Put message ID in vector  
 6E 04 C0 0148 569 ADDL2 #4,(SP) : Advance return address  
 7B 50 E9 0162 570 SGETJPI\_S ITMLST=OPERGETJPI : Get process ID  
 000026A4'EF 03 D0 0165 571 BLBC R0,10\$ : If LBC, error  
 000026AC'EF 01 D0 016C 572 MOVL #3,OPERMSGVEC : Assume WRITE ACTIVE  
 000026B4'EF D4 0173 573 MOVL #1,OPERMSGFAO  
 000026A8'EF 007CA01B 8F D1 0179 574 CLRL OPÉRMSGNAM  
 2D 12 0184 575 CMPL #SYSG\$\_WRITERCUR,OPERMSGID : WRITE CURRENT ?  
 000026A4'EF D6 0186 576 BNEQ 58 : If NEQ, no  
 000026AC'EF D6 018C 577 INCL OPERMSGVEC : Set up WRITE CURRENT  
 000026B4'EF 000026B8'EF 9E 0192 579 INCL OPERMSGFAO  
 000026B8'EF 00000000'EF 9A 019D 580 MOVAB OPERNAMDESC,OPERMSGNAM : Build descriptor  
 000026BC'EF 00000000'EF D0 01A8 581 MOVZBL R10\_INPNAM+NAMSB\_RSL,OPERNAMDESC : Build descriptor  
 01B3 582 SPUTMSG\_S - R10\_INPNAM+NAMSL\_RSA,OPERNAMDESC+4 : Get and format message  
 01B3 583 584 : MSGVEC=OPERMSGVEC, -  
 01B3 585 : ACTRTN=666\$ : If LBC, error  
 13 50 E9 01CA 586 : SSNDOPR\_S MSGBUF=OPERMSG : If LBS, success  
 06 50 E8 01DD 587 : BLBS R0,20\$ : Report error  
 FE1D 30 01E0 588 10\$: BSBW PUTERROR : Force success  
 50 01 D0 01E3 589 20\$: MOVL #1,R0  
 01E6 590 : RSB  
 05 01E6 591 :  
 01E7 592 666\$: .WORD \*M<R2,R3,R4,R5>  
 003C 01E7 593 : MOVQ B4(AP),R0 : Get string descriptor  
 50 04 BC 7D 01E9 594 : ADDL3 #OPCSL\_MS\_TEXT,R0,OPERMSG : Store total operator message size  
 000026D0'EF 50 08 C1 01ED 595 : MOVC3 R0,(R1),OPERMSGTXT : Copy text to operator message buffer  
 61 50 28 01F5 596 : CLRL R0 : Prevent message output to SYSSOUTPUT  
 50 D4 01FD 597 : RET  
 04 01FF 598 :  
 0200 599 :  
 0200 600 : ENDC : End of SYSGEN-specific code

0200 602 .SBTTL BOOSCONFIGALL - Auto-configure all adapters  
 0200 603  
 0200 604 ++ Functional Description:  
 0200 605 BOOSCONFIGALL is called to implement the "AUTOCONFIGURE ALL" command. All standard devices supported by VAX/VMS will be located and connected for use with any necessary drivers being loaded.  
 0200 606  
 0200 607  
 0200 608  
 0200 609  
 0200 610  
 0200 611  
 0200 612  
 0200 613  
 0200 614  
 0200 615  
 0200 616  
 00000089 617 .PSECT NONPAGED\_CODE rd,nwrt,exe,long  
 0089 618  
 OFFC 0089 619 .Entry BOOSCONFIGALL, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>" ; Entry mask  
 0088 620  
 08 00000000'EF 50 00000000'8F E1 008B 621 BBC #EXESV\_NOAUTOCNF,EXESGL\_DEFFLAGS,5\$; do we allow auto configure  
 007C8002 8F D0 0097 622 MOVL #SYSGS\_NOAUTOCNF,RO ;Give them a no autoconfigure error  
 04 009E 623 RET ; and return  
 FF5E' 30 009F 624 5\$: BSBW BOOSLOCK\_GEN ; Lock SYSGEN database  
 07 50 E8 00A2 625 BLBS R0,7\$ ; If no error, continue  
 00000000'EF 16 00A5 626 JSB PU\$ERROR  
 04 00AB 627 RET  
 00AC 628  
 FF51' 30 00AC 629 7\$: BSBW IOCSAUTORESET ; Reset controller characters for device  
 0000013B'EF 5B D4 00AF 630 names  
 5B DD 00B1 631 CLRL R11  
 01 FB 00B3 632 PUSHL R11  
 29 50 E9 00BA 633 10\$: CALLS #1,NEXTADP  
 5B 51 D0 00BD 634 BLBC R0,CONFIG\_EXIT  
 10 18 00C0 635 MOVL R1,R11  
 5B DD 00C2 636 BGEQ 20\$  
 018C'CF 01 FB 00C4 637 PUSHL R11  
 E5 50 E8 00C9 638 CALLS #1,W\$CONFIGADP  
 00000000'EF 16 00CC 639 BLBS R0,10\$  
 50 01 D0 00D2 640 JSB PU\$ERROR  
 00D5 641 RET  
 00D5 642 20\$: MOVL #1,R0  
 09 00000000'EF 0C E1 00D5 643 BBC #BOOCMD\$V\_AUTOLOG,L\*BOOSGL\_CMDOPT,CONFIG\_EXIT ; Branch if not /LOG  
 00002614'EF D4 00DD 644 CLRL BOOSGT\_SAVE\_DEVNAME ; Clear name  
 01A9 30 00E3 645 BSBW AUTOLOG ; Output last line if there is one  
 00E6 646  
 00E6 647  
 00E6 648 CONFIG\_EXIT:  
 50 DD 00E6 649 PUSHL R0  
 FF15' 30 00E8 650 BSBW BOOSUNLOCK\_GEN ; Save status  
 06 50 E8 00EB 651 BLBS R0,35\$ ; Unlock SYSGEN database  
 00000000'EF 16 00EE 652 JSB PU\$ERROR ; If no error, continue  
 50 8ED0 00F4 653 35\$: POPL R0 ; Give error message  
 04 00F7 654 RET ; Restore status  
 00F8 655  
 OFFC 00F8 656 .Entry BOOSCONFIGONE, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>" ; Entry mask  
 00FA 657  
 FF03' 30 00FA 658 BSBW BOOSLOCK\_GEN ; Lock SYSGEN database

07 50	E8	00FD	659	BLBS	RO, SS	If no error, continue
00000000'EF	16	0100	660	JSB	PUTERROR	Give error message
	04	0106	661	RET		
		0107	662			
FEF6'	30	0107	663	SS:	BSBW	IOCSAUTORESET
		010A	664			Reset controller characters for device
1C AC	DD	010A	665	PUSHL	TPASL NUMBER(AP)	device names
63'AF	01	FB	010D	CALLS	#1,B^[OCADP	Set TR number of adapter
D2 50	E9	0111	666	BLBC	RO,CONFIG_EXIT	Locate adapter control block
51	DD	0114	667	PUSHL	R1	Branch if error (NOPRIV)
0D	13	0116	668	BEQL	10\$	Set as argument to CONFIGADP
8C'AF	01	FB	0118	670	CALLS	Done if no adapter
06 50	E8	011C	671	BLBS	#1,B^CONFIGADP	Configure adapter
00000000'EF	16	011F	672	JSB	RO,10\$	Continue if no error
50 01	DD	0125	673	MOVL	PUTERROR	Give error status
B6 00000000'EF	0C	E1	0128	674	BBC	Set success for parse
00002614'EF	D4	0130	675	CLRL	#B00CMDSV AUTOLOG,L^BOOSGL	CMDOPT,CONFIG_EXIT ; Branch if not /LOG
0156	30	0136	676	BSBW	BOOSGT SAVE_DEVNAME	Clear name
AB	11	0139	677	20\$:	BRB	AUTOLOG
		0138	678			Output last line if there is one
		0138	679	NEXTADP:		
	0000	0138	680	.WORD	0	Return next ADP address in RO
		013D	681	\$CMEXEC_S		Null entry mask
	04	0149	682	RET	B^10\$, (AP)	Call real routine in exec mode
		014A	683			
	0000	014A	684	10\$:	.WORD	Null entry mask
51 04 AC	DD	014C	685	MOVL	4(AP),R1	Get current address
06	13	0150	686	BEQL	20\$	0 => start of list
51 04 A1	D0	0152	687	MOVL	ADPSL_LINK(R1),R1	Flink onward
07	11	0156	688	BRB	30\$	
51 00000000'EF	DD	0158	689	20\$:	MOVL	Return head of list
50 01	DD	015F	690	30\$:	MOVL	IOCSGL_APDLIST,R1
	04	0162	691		RET	
		0163	692			
	0163	693	LOCADP:			Return address of ADP for TR number
	0000	0163	694	.WORD	0	
		0165	695	\$CMEXEC_S		Call routine in exec mode
	04	0171	696	RET	B^5\$, (AP)	
		0172	697			
	0000	0172	698	5\$:	.WORD	Null entry mask
51 FFFFFFFC'EF	9E	0174	699	MOVAB	IOCSGL_APDLIST-ADPSL_LINK,R1	Set starting address
51 04 A1	DD	017B	700	10\$:	ADPSL_LINK(R1),R1	Flink onward
07	13	017F	701	BEQL	20\$	Done if at end
OC A1 04 AC	B1	0181	702	[MPW	4(AP),ADPSW_TR(R1)	Is this the specified TR?
F3 12	0186	703		BNEQ	10\$	No, try another
50 01	DD	0188	704	20\$:	MOVL	
	04	0188	705		RET	
		018C	706			
	00FC	018C	707	.Entry	CONFIGADP, ^M<R2,R3,R4,R5,R6,R7>;	Entry mask
		018E	708			
10 000024C4'EF	D4	018E	709	CLRL	BOOSGL RETSAVE	Zap return address for initial call
08 00000000'EF	06	E1	0194	BBC	#B00CMDSV SELECT,L^BOOSGL	CMDOPT,10\$ ; Mutually exclusive - test
07	E1	019C	710	BBC	#B00CMDSV-EXCLUDE,L^BOOSGL	to make sure one bit clear
50 007C808A 8F	DD	01A4	711	MOVL	#SYSGS_CONFQUAL,R0	Conflicting qualifiers
	04	01AB	712	RET		
		01AC	713			
		01AC	714			
01FA'CF	6C	FA	01AC	715	10\$:	CALLG (AP),W^50\$
						; Call configure one device

8 7

50 09 50	E8 01B1	716	BLBS	R0,20\$	: Branch if not done with this adapter	
50 24	B1 01B4	717	CMPW	#55\$_NOPRIV,R0	: Was there a privilege error	
50 03	13 01B7	718	BEQL	15\$	: Yes, branch	
50 01	00 01B9	719	MOVL	#1,R0	: Set success	
	04 01BC	720	15\$:	RET	: and return	
55 0000248C'EF	9E 01BD	722	MOVAB	BOOSAL_ACF,RS	: Set address of arguments describing device	
56 00002460'EF	84 01C4	724	CLRW	ACFSW_MAXUNITS(R5)	: Always use driver specified max units	
06 1C A5	00 01C7	725	MOVL	L^BOOSGL_SELECT,R6	: Get pointer to select list	
0087 06	13 01CE	726	BEQL	35\$	: Branch if null	
D6 50	30 01D0	727	BSBW	SELECT	: Check select/exclude string	
	E9 01D3	728	BLBC	R0,10\$	: Branch if device is not to be configured	
09 00000000'EF	11 0B A5	03	E0 01D6	730	35\$:	BBS #ACFSV_NOLOAD_DB,ACFSB_AFLAG(R5),38\$ ; Branch if not loading database
0C 00A9	E1 01DB	731	BBC #BOOCMD\$V_AUTOLOG,L^BOOSGL_CMDOPT,38\$ ; Branch if not logging			
03 50	30 01E3	732	BSBW AUTOLOG	: Branch to output log		
FE14	E8 01E6	733	BLBS R0,38\$	: Branch if no error		
	30 01E9	734	BSBW PU\$ERROR	: Give error message		
0000'CF	65 FA	01EC	736	38\$:	CALLG (R5) W^IOGEN\$LOADER	: Load database and driver if necessary
BB 50	E8 01F1	737	BLBS R0,10\$	: Branch if no error		
FE09	30 01F4	738	BSBW PU\$ERROR	: Give error message		
FFB2	31 01F7	739	BRW 10\$	: continue loop		
	0000 01FA	740				
	01FC	741	50\$:	.WORD 0		
	0208	742	\$CMKRNL_S B^55\$, (AP)	: Call auto configure in kernel mode		
	04 0209	743	RET			
	0209	744				
50 000024C4'EF	0FFC 0209	745	55\$:	.WORD "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>"	: Get saved return address	
07	00 0208	746	MOVL BOOSGL_RETSAVE,R0	: Branch if one present		
50 00000000'EF	07 12	0212	BNEQ 60\$	: Else use main entry point		
50 58 04 AC	9E 0214	748	MOVAB IOC\$AUTOCONFIG,R0	: Stack call back address		
56 68	DD 021B	749	PUSHL R0	: Get address of ADP		
57 0000248C'EF	00 021D	750	MOVL 4(AP),R8	: Get Configuration register address		
06 50	00 0221	751	MOVL ADPSL_CSR(R8),R6	: Address of configuration control block		
000024C4'EF	9E 0224	752	MOVAB BOOSAC_ACF,R7	: Disable interrupts		
06 50	00 0228	753	SETIPL #31	: Call Auto configuration code		
000024C4'EF	9E 16	022E	JSB 0(SP)+	: Enable interrupts		
06 50	00 0230	754	SETIPL #0	: Save return		
000024C4'EF	BE 00 0233	755	MOVL (SP)+,BOOSGL_RETSAVE	: Continue if another device		
06 50	E8 023A	756	BLBS R0,70\$	: Else clear return		
000024C4'EF	D4 023D	757	CLRL BOOSGL_RETSAVE			
	0243	758				
00 0B A7	03 F1	0243	759	70\$:	BBC #ACFSV_NOLOAD_DB,ACFSB_AFLAG(R7),80\$ ; Branch if loading database	
7E 12 A7	3C 0248	760	MOVZWL ACFSW_CUNIT(R7),-(SP)	: Get unit number		
14 A7	DD 024C	761	PUSHL ACFSL_DEVNAME(R7)	: Get device name		
0108	30 024F	762	BSBW SGN\$GET_DEVICE_LOCK_IODB	: Get device database		
5E 08	C0 0252	763	ADDL2 #8,SP	: Clear stack		
50 01	04 0255	764	RET	: And return		
	0256	765	80\$:	MOVZWL #1,R0	: Set success status	
	04 0259	766	90\$:	RET	: and return	
	025A	767				
	025A	768				
	025A	769	:	SELECT - decide whether current device name is one of those either		
	025A	770	:	specified in /SELECT or /EXCLUDE		
	025A	771	:			
	025A	772	:	Returns: R0 = 1 ==> configure device		

025A 773 : RO = 0 ==> don't configure device

025A 774 :  
025A 775 :  
025A 776 SELECT:  
57 14 A5 D0 025A 777 10\$: MOVL ACFSL\_DEVNAME(R5),R7 : Get pointer to device name  
54 86 9A 025E 778 MOVZBL (R6)+,R4 : Get length of select entry  
54 10 13 0261 779 BEQL 20\$ : End of list, no match  
54 87 91 0263 780 CMPB (R7)+,R4 : Compare with device entry  
54 06 19 0266 781 BLSS 15\$ : Branch if select longer than device  
67 66 54 29 0268 782 CMPC3 R4,(R6),(R7) : Do we have a match?  
13 13 026C 783 BEQL 40\$ : Yes, check SELECT or EXCLUDE  
56 54 C0 026E 784 15\$: ADDL R4,R6 : Advance to next entry in select list  
E7 11 0271 785 BRB 10\$ : And try again  
03 00000000'EF 50 D4 0273 787 20\$: CLRL R0 : Assume don't configure  
07 E1 0275 788 BBC #BOOCMD\$V\_EXCLUDE,BOOSGL : Branch if SELECT  
50 01 D0 027D 789 MOVL #1,R0 : EXCLUDE - configure device  
05 0280 790 30\$: RSB  
0281 791  
03 00000000'EF 50 D4 0281 792 40\$: CLRL R0 : Assume don't configure  
07 E0 0283 793 BBS #BOOCMD\$V\_EXCLUDE,BOOSGL : Branch if EXCLUDE  
50 01 D0 028B 794 MOVL #1,R0 : SELECT - configure device  
05 028E 795 50\$: RSB

028F 797 .SBTTL AUTOLOG - AUTO ALL /LOG formatting

028F 798

028F 799 AUTOLOG::

55 0000248C'EF 9E 028F 800 MOVAB B00\$AL\_ACF,R5 ; Address of configuration control block  
56 14 A5 D0 0296 801 MOVL ACF\$L\_DEVNAME(R5),R6 ; Get address of current device  
00002614'EF 66 57 29 029A 802 MOVZBL (R6)+,R7 ; Get count and addr.  
39 12 02A5 803 CMPC3 R7 (R6),B00\$GT\_SAVE\_DEVNAME ; Compare to previous string  
02A7 804 BNEQ 50\$ ; Branch if new device

02A7 805

02A7 806 SFAO\_S CTR\$TR=CTR\$TR\_AUTOLOG\_UNIT,- ; Format Unit Number  
02A7 807 OUTBUF=OUTBUF,-  
02A7 808 OUTLEN=OUTLEN\_UNIT,-  
02A7 809 P1=ACFSW\_CUNIT(R5)

03 50 E8 02C3 810 BLBS R0,40\$ ; Branch if OK  
0081 31 02C6 811 BRW 100\$ ; Branch if error

2610'CF 260C'CF C0 02C9 813 40\$: ADDL2 W^OUTLEN\_UNIT,W^OUTLEN ; Add to total length  
262C'CF 260C'CF C0 02D0 814 ADDL2 W^OUTLEN\_UNIT,W^OUTBUF+4 ; Add to descriptor  
2628'CF 260C'CF A2 02D7 815 SUBW2 W^OUTLEN\_UNIT,W^OUTBUF ; Subtract from length  
6A 11 02DE 816 BRB 100\$ ; Return with success

2610'CF 21 05 02E0 817 50\$: TSTL W^OUTLEN ; Is this a first call to this routine?  
21 13 02E4 819 BEQL 70\$ ; Branch if yes

262C'CF 2630'CF DE 02E6 821 MOVAL W^OUTBUF\_STR,W^OUTBUF+4 ; reset descriptor  
0000'CF 2610'CF B0 02ED 822 MOVW W^OUTLEN,W^RIOSGW\_OUTLEN ; Length of string  
0000'CF 28 02F4 823 MOVC3 W^RIOSGW\_OUTLEN,-  
2630'CF 02F8 824 W^OUTBUF\_STR,-  
0000'CF 02FB 825 W^RIOSAB\_BUFFER ; Move text into global buffer  
02FE 826

00000000'EF 43 50 16 02FE 827 JSB RIOSOUTPUT\_LINE  
43 50 E9 0304 828 BLBC R0,100\$ ; Branch on error

0307 829

2628'CF 0064 8F B0 0307 830 70\$: MOVW #100,W^OUTBUF ; Set full buffer length  
00002614'EF 66 57 28 030E 831 MOVC3 R7,(R6),B00\$GT\_SAVE\_DEVNAME ; Save new devname  
55 0000248C'EF 9E 0316 832 MOVAB B00\$AL\_ACF,R5 ; Reset R5  
031D 833 SFAO\_S CTR\$TR=CTR\$TR\_AUTOLOG,- ; Format device name  
031D 834 OUTBUF=OUTBUF,-  
031D 835 OUTLEN=OUTLEN,-  
031D 836 P1=ACFSL\_DEVNAME(R5),-  
031D 837 P2=ACFSW\_CUNIT(R5)

262C'CF 2610'CF C0 033C 838 ADDL2 W^OUTLEN,W^OUTBUF+4 ; Add to descriptor  
2628'CF 2610'CF A2 0343 839 SUBW2 W^OUTLEN,W^OUTBUF ; Subtract from length  
05 034A 840 100\$: RSB ; Return with FA0 status  
034B 841

034B 842

034B 844 .SBTTL SGNSGET\_DEVICE - Locate device database

034B 845

034B 846

034B 847 : Inputs:

034B 848 4(SP) - Address of Device name in ascfc format

034B 849 8(SP) - Unit number

034B 850

034B 851 : Outputs:

034B 852 (Any of these are 0 if the data block doesn't exist)

034B 853 ACF\$GL\_DDB - Address of DDB

034B 854 ACF\$GL\_UCB - Address of UCB

034B 855 ACF\$GL\_IDB - Address of IDB

034B 856 ACF\$GL\_CRB - Address of CRB

034B 857 ACF\$GL\_SB - Address of SB

034B 858 ACF\$GL\_LASTDDB - If ACF\$GL\_DDB is non-zero, then equal to that.

034B 859 otherwise, last DDB in DEVLIST

034B 860 R0 = 0 - error

034B 861 = 1 - success

034B 862

034B 863 : Must be called at IPL=0 and KERNEL mode

034B 864

034B 865 .ENABL LSB

034B 866

034B 867 SGNSGET\_DEVICE:: : Entry with IODB MUTEX & raised IPL

034B 868

007C 8F BB 034B 869 PUSHR #^M<R2,R3,R4,R5,R6> : ADDS 20 to offset to input

54 00000000'GF 32 D0 034F 870

034F 871

0356 10 0358 872

0358 873

007C 8F BA 0358 874

05 035C 875

035D 876

035D 877 SGNSGET\_DEVICE\_LOCK\_IODB: : Entry without IODB MUTEX and IPL 0

035D 878

007C 8F BB 035D 879 PUSHR #^M<R2,R3,R4,R5,R6> : ADDS 20 to offset to input

54 00000000'GF D0 0361 880

00000000'GF 16 0368 881

1A 10 036E 882

50 DD 0370 883

54 00000000'GF D0 0372 884

00000000'GF 16 0379 885

037F 886

0382 887

0382 888

007C 8F BA 0382 889

05 0385 890

0389 891

038A 892

2400'CF D4 038A 893 10\$: CLRL W^ACF\$GL\_DDB : INIT TO ZERO

2404'CF D4 038E 894 CLRL W^ACF\$GL\_UCB : INIT TO ZERO

2408'CF D4 0392 895 CLRL W^ACF\$GL\_IDB : INIT TO ZERO

240C'CF D4 0396 896 CLRL W^ACF\$GL\_CRB : INIT TO ZERO

2418'CF D4 039A 897 CLRL W^ACF\$GL\_SB : INIT TO ZERO

039E 898

56 20 AE D0 039E 899 SAVIPL -(SP)

03A1 900 MOVL 32(SP),R6 : SAVE THE CURRENT IPL

:GET ADDR OF DEVICE NAME

55	86	9A	03A5	901	MOVZBL	(R6)+, R5	:GET SIZE OF DEVICE NAME	
7E	55	7D	03AB	902	MOVQ	R5,-(SP)	:FORM DESCRIPTOR	
51	5E	D0	03AB	903	MOVL	SP, R1	:ADDRESS OF DESCRIPTOR	
00000000'GF	16	03AE	904	JSB	G^\$OC\$SEARCHALL	:SEARCH FOR DEVICE		
8E	7C	03B4	905	CLRQ	(SP)+	:GET RID OF TRASH		
2418'CF	53	D0	03B9	907	SETIPL	(SP)+	:RESTORE OLD IPL	
04	12	03BE	908	MOVL	R3, W^ACF\$GL_SB	:STUFF THE SYSTEM BLOCK		
50	D4	03C0	909	BNEQ	20\$	:NO ERROR, CONTINUE		
5E	11	03C2	910	CLRL	R0	:INDICATE ERROR		
			03C4	911	BRB	70\$	:EXIT	
0908 8F	OB	50	E8	03C4	912	20\$: BLBS	:SUCCES - FOUND DEVICE	
		50	B1	03C7	913	CMPW R0, 25\$	:CHECK IF ERROR WAS 'UNIT NOT FOUND'	
		36	12	03CC	914	BNEQ R0, #SS\$_NOSUCHDEV	:IF NOT, PUNT	
		51	D5	03CE	915	TSTL R1	:SEE IF WE GOT BACK A UCB ADDRESS	
		32	12	03D0	916	BNEQ 60\$	:IF NON-ZERO, IS LISTHEAD - NO DDB FOUND	
00002400'EF	52	D0	03D2	917	25\$: MOVL	R2, L^ACF\$GL_DDB	:ADDRESS OF DDB	
54 04	A2	D0	03D9	918	MOVL	DDB\$L_UCB(R2), R4	:GET ADDRESS OF FIRST UCB	
25	13	03DD	919	BEQL	60\$	:IF NO UCB, EXIT WITH OTHER FIELDS ZE\$		
51 24	A4	D0	03DF	920	MOVL	UCB\$L_CRB(R4), R1	:GET ADDR OF CRB	
0000240C'EF	51	D0	03E3	921	MOVL	R1, L^ACF\$GL_CRB	:SAVE	
2408'CF	2C	A1	D0	03EA	922	MOVL	CRB\$L_INTD+VEC\$L_IDB(R1), W^ACF\$GL_IDB	:GET ADDR OF IDB
54 A4 20 AE	B1	03F0	923	03F0	924	30\$: CMPW 32(SP), UCBSW_UNIT(R4)	:IS UCB ALREADY LOADED?	
08	13	03F5	925	03F5	925	BEQL 50\$	:BRANCH IF IT IS	
54 30 A4	D0	03F7	926	03F7	926	40\$: MOVL UCB\$L_LINK(R4), R4	:GET ADDR OF NEXT UCB	
F3	12	03FB	927	03FB	927	BNEQ 30\$	:BR IF THERE IS ONE	
05	11	03FD	928	03FD	928	BRB 60\$	:EXIT WITH UCB = 0	
2404'CF	54	D0	03FF	930	50\$: MOVL	R4, W^ACF\$GL_UCB		
00002410'EF	52	D0	0404	931	60\$: MOVL	R2, ACF\$GL_LASTDDB	:LAST DDB IN LIST AS SEARCHED	
50 00000000'GF	DE	040B	932	040B	932	MOVAL G^\$C\$SSGA_LOCALSB, R0	:GET ADDRESS OF LOCAL SYSTEM BLOCK	
50 53	D1	0412	933	0412	933	CMPL R3, R0	:IS THIS SB LOCAL?	
08	13	0415	934	0415	934	BEQL 65\$	:YES, LEAVE NOW	
18 A3	7D	0417	935	0417	935	MOVQ SB\$B_SYSTEMID(R3), -		
0000244C'EF	041A	936	041A	936	L^BO0\$GQ_CONSYSID	:NO, SET IN THE SYSTEM ID		
50 01	D0	041F	937	041F	937			
	05	0422	938	0422	938	65\$: MOVL #1, R0	:SUCCESS	
		0423	939	0423	939	RSB		
		0423	940	0423	940	.DSABL LSB		
		0423	941	0423	941			
		0423	942	0423	942			

0423 946 .SBTTL Reset routines BOOSRESETLIST and BOOSCONRESET and BOOSMSCP\_RESET  
 0423 945 :  
 0423 946 : BOOSCONRESET - Reset values for connect command  
 0423 947 :  
 0423 948 :  
 00000200 949 .PSECT PAGED\_CODE rd,nowrt,exe,long  
 0200 950 :  
 0000 0200 951 .Entry BOOSCONRESET, ^M<> ; Null entry mask  
 0202 952 :  
 0000245C'EF 00000200'EF 9E 0202 953 MOVAB L^BOOSAB\_PRMBUF,BOOSGL\_NEXTSTR ; Reset for string allocation  
 00002428'EF 01 CE 0200 954 MNGL #1,BOOSGE\_CONCRÉG ; Null control register  
 0000243C'EF 01 CE 0214 955 MNGL #1,BOOSGL\_CONAUNIT ; Null adapter unit  
 00002434'EF 01 CE 021B 956 MNGL #1,BOOSGL\_CONVECT ; Null vector  
 00002438'EF 01 DO 0222 957 MOVL #1,BOOSGL\_CONNUMV ; Default number of vectors  
 00002424'EF 02 CE 0229 958 MNGL #2,BOOSGL\_CONADP ; Invalidate adapter TR value  
 00002440'EF D4 0230 959 CLRL BOOSGL\_CONDEV ; Clear device name pointer  
 00002444'EF D4 0236 960 CLRL BOOSGL\_CONDRV ; and driver name pointer  
 00002448'EF D4 023C 961 CLRL BOOSGL\_CONUNITS ; and maximum units  
 0000244C'EF 7C 0242 962 CLRQ BOOSGO\_CONSYSID ; and system id  
 00002458'EF D4 0248 963 CLRL BOOSGL\_CONFLAGS ; and flags  
 00002430'EF 01 DO 024E 964 MOVL #1,L^BOOSGL\_CONNUMU ; Set number of units to 1  
 0000241C'EF D4 0255 965 CLRL BOOSGL\_COMBO\_VECTOR\_OFFSET ; Set vector offset from combo vectors to  
 00002420'EF D4 025B 966 CLRL BOOSGL\_COMBO\_CSR\_OFFSET ; Set CSR offset from combo CSR to 0  
 04 0261 967 RET ; Return  
 0262 968 :  
 0262 969 : BOOSRESETLIST - Reset select list values  
 0262 970 :  
 0000 0262 971 .Entry BOOSRESETLIST, ^M<> ; Null entry mask  
 0264 972 :  
 00002460'EF D4 0264 973 CLRL BOOSGL\_SELECT ; Zap select list pointer  
 00000200'EF 9E 026A 974 MOVAB BOOSAB\_PRMBUF,BOOSGL\_NEXTSTR ; Set next string address  
 00002614'EF D4 0275 975 CLRL BOOSGT\_SAVE\_DÉVNAME ; Clear autolog string  
 00002610'EF D4 027B 976 CLRL OUTLEN ; Clear autolog output size  
 0000262C'EF 00002630'EF DE 0281 977 MOVAL OUTBUF\_STR,OUTBUF+4 ; Set address in descriptor of block  
 00002497'EF 94 028C 978 CLRB BOOSAL\_ACF+ACFSB\_AFLAG ; Clear ACF flags  
 00002430'EF 01 DO 0292 979 MOVL #1,L^BOOSGL\_CONNUMU ; Set number of units to 1  
 04 0299 980 RET ; and return  
 029A 981 :  
 029A 982 : BOOSMSCP\_RESET - Reset the MSCP server initialization argument list  
 029A 983 :  
 029A 984 :  
 003C 029A 985 .Entry BOOSMSCP\_RESET, ^M<R2,R3,R4,R5> ; Entry mask  
 029C 986 :  
 FF5F CF 00 FB 029C 987 CALLS #0,BOOSCONRESET ; Reset the connect command globals  
 50 0084 8F 3C 02A1 988 MOVZWL #SSS\_DEVOFFLINE,R0 ; Assume error  
 00000000'GF D5 02A6 989 TSTL G^SCS\$GL\_CDL ; SCS loaded?  
 2C 13 02AC 990 BEQL 10\$ ; If eql no, error  
 50 02C4 8F 3C 02AE 991 MOVZWL #SSS\_DEVACTIVE,R0 ; Assume error  
 03 00 02 F0 02B3 992 INSV #2,#0,#3,R0 ; Set E class error status  
 00000000'GF D5 02B8 993 TSTL G^SCS\$GL\_MSCP ; If neq already loaded  
 1A 12 02BE 994 BNEQ 10\$ ; Exit with error  
 00002444'GF 000025EC'EF DE 02C0 995 MOVAL MSCP\_NAME,G^BOOSGL\_CONDRV ; Set pointer to MSCP server name  
 000025BC'GF 0000258C'EF 30 28 02CB 996 MOVC3 #MSCP\_ARG\_LIST\_SIZE ; Set up default argument list for  
 50 01 DO 02CD 997 MOVL MSCP\_ARG\_LIST,G^BOOSGL\_LOAD\_ARGS ; MSCP server init routine  
 04 02D7 998 #1,R0 ; Set success  
 02DB 1000 999 10\$: RET ; and return

50 20 AC 00	02DB 0000	1001 1002	02DB 02DB	1003 : BOOSMSCP_ARG - Load MSCP arguments	
1C AC 00	02DD 02E1	1004 .Entry BOOSMSCP_ARG, "M<>"	02DD 02E4	1005 : Entry mask	
50 01 00	02EA 02ED	1006 MOVL TPASL_PARAM(AP), R0	02EE 02EE	1007 MOVL TPASL_NUMBER(AP), -	
	04	1008 G^BOOSGL_LOAD_ARGS[R0]	02EE 02EE	1009 MOVL #1, R0	
		1010 RET	02EE 02EE	1011	
		1012	02EE 02EE	1013 :	
		1014 : BOOSMAKLIST - Make a select list entry	02EE 02EE	1015 :	
	007C	1016 .Entry BOOSMAKLIST, "M<R2,R3,R4,R5,R6>"	02F0	1017 : Entry mask	
6 0000245C'EF	00002460'EF	02F0 02F7	1018 1019	MOVL L^BOOSGL_NEXTSTR, R6	1019 : Get pointer to next available string space
07	07	02FD 12	1020	TSTL L^BOOSGL_SELECT	1020 : Is selection pointer already set
0002460'EF	56	02FF 00	1021	BNEQ 10S	1021 : Yes, continue to add entry
50 10 AC	00	0306 030A	1022 10S:	MOVL R6, L^BOOSGL_SELECT	1022 : Else set pointer to first select entry
86 50	90	030A 030D	1023	MOVL TPASL_TOKENCNT(AP), R0	1023 : Get string length
66 14 BC	50	030D 28	1024	MOVB R0, (R6)+	1024 : Set count for string
000245C'EF	63	0312 94	1025	MOVC3 R0, @TPASL_TOKENPTR(AP), (R6) ; Copy string body	1025 : Mark end of list
50 01	00	0314 031B	1026 1027	CLRB (R3)	1026 : Save next string address
	04	031B 031E	1028	MOVL R3, L^BOOSGL_NEXTSTR	1028 : Set success status
				RET	

00002424'EF	1C AC	0000	031F	1030	.SBTTL BOOSCONADP - Set connect adapter number
		00	031F	1031	.Entry BOOSCONADP, ^M<>
		04	0321	1032	MOVL TPASL_NUMBER(AP),L^BOOSGL_CONADP ; Set adapter number
			0329	1033	RET ; and return
00002424'EF	01	0000	032A	1034	.Entry BOOSCONNADP, ^M<>
		CE	032C	1035	MNEGL #1,L^BOOSGL_CONADP ; Connect with null adapter
		04	0333	1036	RET ; Clear adapter number
			0334	1037	;
0000241C'EF	1C AC	0000	0334	1038	.Entry BOOSCONVECOFFSET, ^M<>
		00	0336	1039	MOVL TPASL_NUMBER(AP),-L^BOOSGL_COMBO_VECTOR_OFFSET ; Offset from start of combo vectors
		04	0339	1040	RET ; Set offset value
			033E	1041	;
00002420'EF	1C AC	0000	033F	1042	.Entry BOOSCONCSROFFSET, ^M<>
		00	0341	1043	MOVL TPASL_NUMBER(AP),-L^BOOSGL_COMBO_CSR_OFFSET ; Offset from start of combo CSRs
		04	0344	1044	RET ; Set offset value
			0349	1045	;
00002428'EF	1C AC	0D 00	0000	034A	.Entry BOOSCONCREG, ^M<>
		EF	034C	1046	EXTZV #0,#13,TPASL_NUMBER(AP),L^BOOSGL_CONCREG ; Control register address
		04	0356	1047	RET ; and return
			0357	1048	;
1C AC	FFFFFE03 8F	0000	0357	1049	.Entry BOOSCONCVEC, ^M<>
00002434'EF	CB	0359	1050	MOVL BICL3 #^xFFFFE03,TPASL_NUMBER(AP),L^BOOSGL_CONVECT ; Set controller vector	
		0361	1051	RET ; and return	
		04	0366	1052	;
00002438'EF	1C AC	0000	0367	1053	.Entry BOOSCONCNUM, ^M<>
		00	0369	1054	MOVL TPASL_NUMBER(AP),L^BOOSGL_CONNUMV ; Number of vectors
		04	0371	1055	RET ; and return
			0372	1056	;
0000243C'EF	1C AC	0000	0372	1057	.Entry BOOSCONAUNIT, ^M<>
		00	0374	1058	MOVL TPASL_NUMBER(AP),L^BOOSGL_CONAUNIT ; Adapter unit number
		04	037C	1059	RET ; and return
			037D	1060	;
		007C	037D	1061	.Entry BOOSCONDVRNAME, ^M<R2,R3,R4,R5,R6> ; Entry mask (R2-R6)
			037F	1062	;
56 0000245C'EF	DO	037F	1063	MOVL 1068 L^BOOSGL_NEXTSTR,R6 ; Address of next string storage	
00002444'EF	56	DO	0386	1064	MOVL R6,BOOSGL_CONDVR ; Save pointer to driver name
	86 10 AC	90	038D	1065	MOVB TPASL_TOKENCNT(AP),(R6)+ ; Set count for string
0000245C'EF	56 10 AC	C1	0391	1066	ADDL3 TPASL_TOKENCNT(AP),R6,BOOSGL_NEXTSTR ; Mark string allocated
66 14 BC	10 AC	28	039A	1067	MOVC3 TPASL_TOKENCNT(AP),@TPASL_TOKENPTR(AP),(R6) ; Copy string
	50 01	DO	03A0	1068	MOVL #1,R0 ; and return success
		04	03A3	1069	RET ;
			03A4	1070	;
			00FC	1071	.Entry BOOSDEVNAME, ^M<R2,R3,R4,R5,R6,R7> ; Device name/unit
			03A6	1072	;
56 0000245C'EF	DO	03A6	1073	MOVL BOOSGL_NEXTSTR,R6 ; Get pointer to next available string	
	56 14 AC	DO	03AD	1074	MOVL TPASL_TOKENPTR(AP),R6 ; Get pointer to string
	53 10 AC	DO	03B1	1075	MOVL TPASL_TOKENCNT(AP),R3 ; And number of characters
00002588'EF	D4	03B5	1076	CLRL FULL_NAME_PTR ; Initialize full device name	
	57 86	9E	03B8	1077	MOVAB (R6),R7 ; Save pointer
	64 53	26	03B8	1078	LOCC #^A/\$/,R3,(R4) ; Find any possible '\$'
	22	3A	03B8	1079	BEQL 8S ; None, just continue
00002588'EF	57	DO	03C4	1080	MOVL R7,FULL_NAME_PTR ; Store pointer

55	53	50	C3	03CB	1086	SUBL3	R0,R3,R5	: Number of characters in node
67	55	01	81	03CF	1087	ADD3	#1,R5 (R7)	: Set in size (incl '\$')
		03	BB	03D3	1088	PUSHR	#^M<R0,R1>	: Save registers
66	64	53	28	03D5	1089	MOV3	R3,(R4),(R6)	: Copy full string
		56	DO	03D9	1090	MOVL	R3,R6	: Save ending address
53	50	01	BA	03DC	1091	POPR	#^M<R0,R1>	: Restore registers
54	51	01	C3	03DE	1092	SUBL3	#1,R0,R3	: Number of characters left
		86	C1	03E2	1093	ADD3	#1,R1,R4	: Pointer to string
55	55	86	9E	03E6	1094	8\$: MOVAB	(R6)+,R5	: Save pointer to count byte
		65	94	03E9	1095	CLRB	(R5)	: Initialize count to zero
52	52	D4	03EB	1096	CLRL	R2	: Initialize unit accumulator	
50	84	9A	03ED	1097	10\$: MOVZBL	(R4)+,R0	: Get a character from device name	
30	50	91	03F0	1098	CMPB	R0,#^A/0/	: And check for a digit	
39	50	1F	03F3	1099	BLSSU	20\$	: Branch if not	
		91	03F5	1100	CMPB	R0,#^A/9/	: Final check for digit	
86	50	90	03FA	1102	20\$: BLEQU	40\$	: Yes it is	
		65	96	03FD	1103	MOVB	R0,(R6)+	: Part of device name
		67	96	03FF	1104	INC3	(R5)	: Increase count
E9	53	F5	0401	1105	INC3	(R7)	: Including nodename	
		16	11	0404	1106	SOBGTR	R3,10\$	: Continue
50	84	9A	0406	1107	30\$: MOVZBL	(R4)+,R0	: Get another digit	
50	30	C2	0409	1108	40\$: SUBL	#^A/0/,R0	: Get value	
52	0A	C4	040C	1109	MULL	#10,R2	: Scale accumulator before adding digit	
2F	19	040F	1110	BLSS	60\$	: Error		
50	09	D1	0411	1111	CMP3	#9,R0	: Check for numeric	
		2A	19	0414	1112	BLSS	60\$	: Error if not
52	50	CO	0416	1113	ADD3	R0,R2	: And add new digit	
EA	53	F5	0419	1114	SOBGTR	R3,30\$	: Continue for entire unit number	
0000245C'EF		DO	041C	1115	50\$: MOVL	R6,BOOSGL_NEXTSTR	: Save updated string pointer	
0000242C'EF		DO	0423	1116	MOVL	R2,BOOSGL_CONCUNIT	: Set unit number	
0000243C'EF		DO	042A	1117	MOVL	R2,BOOSGL_CONAUNIT	: Assume same for adapter unit	
00002440'EF		DO	0431	1118	MOVL	R5,BOOSGL_CONDEV	: Save device name pointer	
		65	95	0438	1119	TSTB	(R5)	: Must not be null device name
		04	13	043A	1120	BEQL	60\$	: Error if so
50	01	DO	043C	1121	MOVL	#1,R0	: Return success	
		04	043F	1122	RET		: and return	
50	D4	0440	1123	60\$: CLRL	R0	: Return error status		
		04	0442	1124	RET		:	
			0443	1125				
00002448'EF	1C AC	0000	0443	1126	.Entry BOOSCONUNITS, "M<>		: Maximum units to be connected	
		DO	0445	1127	MOVL TPASL_NUMBER(AP),L"BOOSGL_CONUNITS	: Set maximum units		
		04	044D	1128	RET	: and return		
			044E	1129				
0000244C'EF	1C AC	0000	044E	1130	.Entry BOOSCONSYSID_LOW, "M<>	: System ID		
		DO	0450	1131	MOVL TPASE_NUMBER(AP), -			
		04	0458	1132	L"BOOSGL_CONSYSID	: Set System ID (low longword)		
			0458	1133	RET	: and return		
			0459	1134				
00002450'EF	1C AC	0000	0459	1135	.Entry BOOSCONSYSID_HIGH, "M<>	: System ID		
		DO	045B	1136	MOVL TPASC_NUMBER(AP), -			
		04	0463	1137	L"BOOSGL_CONSYSID+4	: Set System ID (high longword)		
			0463	1138	RET	: and return		
			0464	1139				
			0464	1140	.Entry BOOSCONSOLE, "M<>	: Connect console block stor. device		
00002424'EF	01 CE	0466	1142	MNEGL #1,L"BOOSGL_CONADP		: No adapter		

0000243C'EF	01	DO	046D	1143	MOVL	#1,L^BOOSGL_CONAUNIT	; Set adapter unit = 1 (not used)
0000242C'FF	01	DO	0474	1144	MOVL	#1,L^BOOSGL_CONCUNIT	; Set unit = 1
00002440'EF	000024D6'FF	9E	047B	1145	MOVAB	L^CONSNNAME,[^BOOSGL_CONDEV	; Set device name pointer
00002438'EF	02	DO	0486	1146	MOVL	#2,L^BOOSGL_CONNUMV	; Set 2 vectors
00002428'FF		D4	048D	1147	CLRL	L^BOOSGL_CONCREG	; No control register
00002430'EF	01	DO	0493	1148	MOVL	#1,L^BOOSGL_CONNUMU	; Set number of units to 1
00002448'EF	02	DO	049A	1149	MOVL	#2,L^BOOSGL_CONUNITS	; Set max units to 2 (OPAO is 1st unit)
00000000'EF		16	04A1	1150	JSB	IOGEN\$CONSOLE	; Do cpu dependent stuff
50	01	DO	04A7	1151	MOVL	#1,R0	
		04	04AA	1152	RET		

04AB 1154 .SBTTL BOOSCONNECT - Connect specified device and load driver  
 04AB 1155  
 04AB 1156 : BOOSCONNECT - Allows a single device to be introduced, appropriate data  
 04AB 1157 structures allocated and initialized, the driver loaded if  
 04AB 1158 required and the controller and device initialized.  
 04AB 1159  
 OFFC 04AB 1160 .Entry BOOSCONNECT, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>" :  
 04AD 1161  
 FB50' 30 04AD 1162 BSBW BOOSLOCK\_GEN : Lock SYSGEN database  
 7F 50 E9 04B0 1163 BLBC R0,70\$ : If error, exit  
 04B3 1164  
 04B3 1165  
 04B3 1166 : Value of BOOSGL\_CONADP  
 04B3 1167  
 04B3 1168 0 or greater => /ADAPTER=n specified  
 04B3 1169 -1 => /NOADAPTER specified  
 04B3 1170 -2 => not specified  
 04B3 1171  
 04B3 1172  
 00002424'EF 04B3 1173 5\$: TSTL L^BOOSGL\_CONADP : Has an adapter been specified?  
 44 18 04B9 1174 BGEQ 20\$ : If so, branch  
 00002424'EF FFFFFFFF 04B8 1175 CMPL #-1,L^BOOSGL\_CONADP : Null adapter?  
 28 13 04C6 1176 BEQL 10\$ : Branch if yes  
 00002424'EF FFFFFFFE 04C8 1177 CMPL #-2,L^BOOSGL\_CONADP : None specified in CONNECT?  
 09 13 04D3 1178 BEQL 7\$ : Figure it out from the database  
 50 007C80D2 04D5 1179 MOVL #SYSGS\_NOADAPTER,R0 : Set no adapter specified error  
 45 11 04DC 1180 BRB 60\$ : exit  
 04DE 1181  
 35 50 04DE 1182 7\$: SCMKRNL\_S W^CONN\_ADAP : Get adapter number from I/O database  
 C3 11 04EB 1183 BLBC R0,60\$ : Exit with error  
 04EE 1184 BRB 58 : Dispatch now on adapter type  
 04FO 1185  
 00 11 04FO 1186 10\$: SCMKRNL\_S W^CONNADP : Change mode to see data base  
 04FD 1187 BRB 30\$ : Continue  
 04FF 1188  
 04FF 1189 20\$: SCMKRNL\_S W^CONNECT : Change mode to see data base  
 00000000'EF 0000248C'EF 050C 1190 30\$: BLBC R0,40\$ : Error occurred  
 0E 50 E9 050F 1191 CALLG L^BOOSAL\_ACF,IOGENSLOADER : Load database and driver  
 03 50 F8 051A 1192 BLBS R0,50\$ : Branch if success  
 FAE0' 30 051D 1193 40\$: BSBW PUTERROR : Give error message  
 50 01 D0 0520 1194 50\$: MOVL #1,R0 : Set success for parser  
 50 DD 0523 1195 60\$: PUSHL R0 : Save error status  
 FAD8' 30 0525 1196 BSBW BOOSUNLOCK\_GEN : Unlock SYSGEN database  
 03 50 E8 0528 1197 BLBS R0,65\$ : If no error, continue  
 FAD2' 30 052B 1198 BSBW PUTERROR : Give error message  
 50 8ED0 052E 1199 65\$: POPL R0 : Restore status  
 04 0531 1200 RET :  
 FACB' 30 0532 1201 70\$: BSBW PUTERROR : Give error message  
 04 0535 1202 RET :  
 0536 1203 : Local routine to get adapter number from I/O database  
 0536 1204 : Must be called by a CMKRNL since SGNSGET\_DEVICE must be called  
 0536 1205 : in Kernel mode.  
 0536 1206 :  
 0536 1207 :  
 0000 0536 1208 .Entry CONN\_ADAP, "M<" :  
 0538 1209 MOVZWL L^BOOSGL\_CONCUNIT,-(SP) ; Unit number

00002440'EF	DD	053F	1211	PUSHL	L^BOOSGL_CONDEV	: Device name	
0000035D'EF	16	0545	1212	JSB	SGNSGET_DEVICE_LOCK_IODB	: Get device data base addresses	
5E 08	CO	054B	1213	ADDL2	#8,SP	: Pop off input parameters	
50 00002408'EF	DO	054E	1215	MOVL	L^ACFSGL_IDB,RO	: Address of IDB	
09	12	0555	1216	BNEQ	58	: Error if zero	
50 007C80D2 8F	DO	0557	1217	MOVL	#SYSGS_NOADAPTER,RO	: Set no adapter specified error	
18	11	055E	1218	BRB	20\$	: Branch to exit	
00002424'EF	01	CE	0560	1220	5\$: MNEGL	#1, L^BOOSGL_CONADP	: Assume null adapter
50 14 A0	DO	0567	1221	MOVL	IDBSL_ADP(RO),RO	: Address of ADP block	
08	13	056B	1222	BEQL	10\$	: Null adapter if zero	
00002424'EF	0C A0	3C	056D	1223	MOVZWL	ADPSW_TR(RO),L^BOOSGL_CONADP	: Set adapter number
			0575	1224			
50 01	DO	0575	1225	10\$:	MOVL	#1,RO	: Set success
	04	0578	1226	20\$:	RET		: Return
		0579	1227				

0579 1229 .ENABL LSB  
0579 1230 ; Connect with null adapter  
0579 1231  
OFFC 0579 1232 .Entry CONNLADP, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>  
057B 1233  
50 00002588'EF DD 057B 1234 MOVL L^FULL\_NAME\_PTR, R0 : Try full device name  
07 12 0582 1235 BNEQ 5\$ : Good, continue  
50 00002440'EF DD 0584 1236 MOVL L^BOOSGL\_CONDEV, R0 : Use normal name  
5A 0000248C'EF 9E 058B 1237 5\$: MOVAB L^BOOSAL\_ACF, R10 : Address ACF  
6A D4 0592 1238 CLRL ACFSL\_ADAPTER(R10) : Set no adapter  
04 AA D4 0594 1239 CLRL ACFSL\_CONFIGREG(R10) : Set address of config reg  
08 AA B4 0597 1240 CLRW ACFSW\_AVECTOR(R10) : Set SCB offset for adapter  
01 E1 059A 1241 BBC #ACFSV\_CRBBLT,- : Br. if CRB built flag is clear  
4C 00002458'EF 059C 1242 BOOSGL\_CONFLAGS, 17\$  
6A 00002454'EF DD 05A2 1243 MOVL BOOSGL\_CONCRB, ACFSL\_ADAPTER(R10) : Store CRB address  
43 11 05A9 1244 BRB 17\$ ; Join common code  
05AB 1245  
OFFC 05AB 1246 .Entry CONNECT, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>" ;  
05AD 1247  
5B FFFFFFFC'GF 9E 05AD 1248 MOVAB G^IOCSGL\_APPLIST-ADPSL\_LINK, R11 ; Get address of adapter list  
5B 04 AB DD 05B4 1249 10\$: MOVL ADPSL\_LINK(R11), R11 : Flink onward through adapter list  
08 12 05B8 1250 BNEQ 15\$ : Continue if another adapter  
50 007C80BA BF DD 05B8A 1251 MOVL #SYSGS\_INVADAP, R0 : Set invalid adapter error  
04 05C1 1252 RET ; Return  
00002424'EF 0C AB B1 05C2 1254 15\$: CMPW ADPSW\_TR(R11), L^BOOSGL\_CONADP ; Is this the specified TR?  
5A 0000248C'EF E8 12 05CA 1255 BNEQ 10\$ : No, try another  
6A 5B DO 05CC 1256 MOVAB L^BOOSAL\_ACF, R10 : Get address of ACF  
04 AA 6B DO 05D3 1257 MOVL R11, ACFSL\_ADAPTER(R10) : Set address of ADP  
50 1C AB 00000000'GF C3 05DA 1258 MOVL ADPSL\_CSR(R11), ACFSL\_CONFIGREG(R10) : Set address of config reg  
08 AA 50 B0 05E3 1260 SUBL3 G^EXESGL\_SCB, - : Calculate offset into SCB of  
50 00002440'EF DD 05E3 1261 MOVW ADPSL\_AVECTOR(R11), R0 : adapter's interrupt vectors.  
05EE 1262 MOVL R0, ACFSW\_AVECTOR(R10) : Store offset in ACF.  
05EE 1263 L^BOOSGL\_CONDEV, R0 ; Device name  
14 AA 00002440'EF DD 05EE 1264 17\$: MOVL BOOSGL\_CONDEV, ACFSL\_DEVNAME(R10); Set pointer to device name  
05F6 1265  
05F6 1266  
05F6 1267 ; Now try to get driver name from DDB if it exists and load BOOSGL\_CONSYSID  
05F6 1268 ; if HSC device.  
05F6 1269  
7E 0000242C'EF 3C 05F6 1270 MOVZWL L^BOOSGL\_CONCUNIT,-(SP) ; Unit number  
50 DD 05FD 1271 PUSHL R0 ; Device name  
0000035D'EF 16 05FF 1272 JSB SGN\$GET\_DEVICE\_LOCK\_IODDB ; Get device data base addresses  
5E 08 C0 0605 1273 ADDL2 #8, SP ; Pop off input parameters  
08 50 E8 0608 1274 BLBS R0, 20\$ ; All okay  
50 007C9010 BF DD 0608 1275 MOVL #SYSGS\_NODEV, R0 ; Set error code - "Device not known"  
04 0612 1276 RET ; Leave  
0613 1277  
00 00002458'EF 05 E2 0613 1278 20\$: BBSS #ACFSV\_GETDONE,-  
0615 1279 L^BOOSGL\_CONFLAGS, 21\$ ; Notify LOADER that GET was done  
061B 1280  
18 AA 00002444'FF DD 061B 1281 21\$: MOVL BOOSGL\_CONDRV, ACFSL\_DRVNAME(R10); And driver name  
31 14 0623 1282 BGTR 30\$ ; Branch if driver specified  
51 00002400'EF DD 0625 1283 MOVL ACFSL\_DDB, R1 ; DDB address  
07 13 062C 1284 BEQL 25\$ ; Branch if none  
18 AA 24 A1 DE 062E 1285 MOVAL DDBST\_DRVNAME(R1), ACFSL\_DRVNAME(R10) ; Address from DDB

8 8

21	11	0633	1286	BRB	30\$	; Branch around name hackery	
56	0000245C'EF	D0	0635	1288	25\$:	MOVL L^BOOSGL_NEXTSTR,R6	; Get address of next free space
18 AA	56	D0	063C	1289		MOVL R6,ACFSL_DRVNAME(R10)	; Set as driver name address
86	08	90	0640	1290		MOVB #8,(R6)+	; Set count for string
51	14 AA	D0	0643	1291		MOVQ #^A/ DRIVER/, (R6)	; Set driver suffix
66	01 A1	B0	064E	1292		MOVL ACFSL_DEVNAME(R10),R1	; Pointer to device name
			0652	1293		MOVW 1(R1),(R6)	; Form default driver name
0A AA	0000243C'EF	90	0656	1294	30\$:	MOVB BOOSGL_CONAUNIT,ACFSB_AUNIT(R10)	; Set adapter unit
21 AA	00002430'EF	90	065E	1295		MOVB L^BOOSGL_CONNUMU,ACFSB_NUMUNIT(R10)	; Store number of units to configure
0B AA	00002458'EF	90	0666	1297		MOVBL BOOSGL_CONFLAGS,ACFSB_AFLAG(R10)	; Store flags
0000241C'EF	00002434'EF	A1	066E	1299		ADDW3 BOOSGL_CONVECT,BOOSGL_COMBO_VECTOR_OFFSET,-;	
10 AA			0679	1300		ACFSW_CVECTOR(R10)	; Set vector address
50	0000241C'EF	08 02	EF	1301		EXTZV #2,#8,BOOSGL_COMBO_VECTOR_OFFSET,R0	; Save vector offset in longwords
1F AA	50	90	0684	1302		MOVBL R0,ACFSB_COMBO_VECTOR_OFFSET(R10);	
			0688	1303			
			0688	1304			
			0688	1305			
			0688	1306			
0C AA	0000244C'EF	D5	0688	1307		TSTL BOOSGQ_CONSYSID	; See if SYSIDLOW was specified
	0A	13	068E	1308		BEQL 40\$	; Branch if not
	0000244C'EF	9E	0690	1309		MOVAB BOOSGQ_CONSYSID,-	
			0698	1310		ACFSL CONTRLREG(R10)	; Set system id address
	22	11	0698	1311		BRB 50\$	; Branch
			069A	1312			
			069A	1313			
			069A	1314			
00002428'EF	00001000 8F	C1	069A	1315	40\$:	ADDL3 #UBA_IOPBASE, -	
OC AA			06A5	1316		BOOSGL_CONCREG, -	
			06A7	1317		ACFSL CONTRLREG(R10)	; control register address
0C AA	04 AA	C0	06A7	1318		ADDL ACFSL_CONFIGREG(R10), -	
			06AC	1319		ACFSL CONTRLREG(R10)	
	00002420'EF	C0	06AC	1320		ADDL BOOSGE_COMBO_CSR_OFFSET,-;	Add adapter va base
OC AA			06B2	1321		ACFSL CONTRLREG(R10)	
	00002420'EF	8E	06B4	1322		MNEG8 BOOSGE_COMBO_CSR_OFFSET,-;	Add offset to get true CSR address
20 AA			06BA	1323		ACFSL CONTRLREG(R10)	
			06BC	1324		BOOSGE_COMBO_CSR_OFFSET,-;	Calculate offset back to CSR start
12 AA	0000242C'EF	B0	06BC	1325	50\$:	ACFSL_COMBO_CSR_OFFSET(R10); Save offset	
1C AA	00002448'EF	B0	06C4	1326		MOVW BOOSGL_CONCUNIT, -	
1E AA	00002438'EF	90	06CC	1327		MOVW ACFSL_CUNIT(R10)	; Set controller unit number
	50 01	D0	06CC	1328		MOVW BOOSGE_CONUNITS, -	
			06D4	1329		ACFSL_MAXUNITS(R10)	; Set maximum units
			06D4	1330		MOVW BOOSGE_CONNUMV, -	
			06D8	1331	55\$:	ACFSL_CNUMVEC(R10)	
			06D8	1332		#1,R0	; Set count of vectors
			06D8	1333		RET	; Set success
			06D8	1334			
			06D8	1335			
						.DSABL LSB	

06D8 1337 .SBTTL BOOSLOAD - Load a driver or misc code if not already loaded  
06D8 1338 :  
06D8 1339 : BOOSLOAD - Loads the driver or misc code if not already loaded.  
06D8 1340 :  
OFFC 06D8 1341 .Entry BOOSLOAD, "M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>  
06DA 1342 :  
52 D4 06DA 1343 CLRL R2 : Clear reload flag  
05 11 06DC 1344 BRB LOADRV : And merge with common code

06DE 1346 .SBTTL BOOSRELOAD - Reload a specified driver  
 06DE 1347 :  
 06DE 1348 : BOOSRELOAD - Reloads the specified driver replacing any existing copy  
 06DE 1349 : unless there are busy units requiring the driver that would  
 06DE 1350 : be replaced.  
 06DE 1351 :  
 OFFC 06DE 1352 .Entry BOOSRELOAD, "MCR2,R3,R4,R5,R6,R7,R8,R9,R10,R11" ;  
 S2 01 90 06E0 1353 :  
 06E3 1354 MOVB #ACFSM\_RELOAD,R2 ; Set flag to force reload  
 06E5 1355 LOADRV:  
 06E3 1356 :  
 06E3 1357 : The first block of the file will be read to determine if it is a driver or  
 06E3 1358 : misc code by looking at the type field.  
 06E3 1359 :  
 00002444'EF DD 06E3 1360 PUSHL BOOSGL\_CONDRV ; File name string  
 6E D6 06E9 1361 INCL (SP) ; Get past the count  
 7E 00002444'FF 9A 06EB 1362 MOVZBL @BOOSGL\_CONDRV,-(SP) ; Length of file name  
 57 SE D0 06F2 1363 MOVL SP,R7 ; Address of descriptor for file name  
 F908' 30 06F5 1364 BSBW BOOSFILEOPEN ; Open the file (default SYSSYSTEM:.EXE)  
 SE 50 E9 06FB 1365 BLBC R0,40\$ ; Error  
 56 00002200'EF 9E 06FE 1366 ADDL #8,SP ; Clean up stack  
 58 02 D0 0705 1368 MOVL #2,R8 ; Buffer for file read  
 59 01 D0 0708 1369 MOVL #1,R9 ; First block after image header  
 F8F2' 30 0708 1370 BSBW BOOSREADFILE ; One page  
 48 50 E9 070E 1371 BLBC R0,40\$ ; Error  
 F8EC' 30 0711 1372 BSBW BOOSFILECLOSE ; Close the currently open file  
 42 50 E9 0714 1373 BLBC R0,40\$ ; Error  
 0000220A'EF 91 0717 1374 CMPB BOOSAB\_LOADBUF+SLVSB\_TYPE,- ; Error  
 62 8F D0 071D 1375 #DYNSC\_LOADCODE ; Check for misc code  
 3C 13 071F 1376 BEQL LOADCODE  
 F8DC' 30 0721 1377 BSBW BOOSLOCK\_GEN ; Lock SYSGEN database  
 32 50 E9 0724 1378 BLBC R0,40\$ ; If lbc, didn't get lock  
 18 AA 0000248C'EF 9E 0727 1379 MOVAB L^BOOSAL\_ACF,R10 ; Get base address for ACF block  
 0B AA D0 072E 1380 MOVL BOOSGL\_CONDRV,ACFSL\_DRVNAME(R10)  
 52 90 0736 1381 MOVB R2,ACFSB\_AFLAG(R10) ; Set flags for load or reload  
 14 AA D4 073A 1382 CLRL ACFSL\_DEVNAME(R10) ; No device name  
 00000000'EF 6A FA 073D 1383 CALLG (R10),L^IOGENSLADER ; Load requested driver  
 03 50 E8 0744 1384 BLBS R0,20\$ ; Continue if no error  
 F886' 30 0747 1385 BSBW PUTERROR ; Give error message  
 50 DD 074A 1386 20\$: PUSHL R0 ; Save status  
 F881' 30 074C 1387 BSBW BOOSUNLOCK\_GEN ; Unlock SYSGEN database  
 03 50 E8 074F 1388 BLBS R0,30\$ ; If no error, continue  
 F8AB' 30 0752 1389 BSBW PUTERROR ; Give error message  
 50 8ED0 0755 1390 30\$: POPL R0 ; Restore status  
 04 0758 1391 RET ; Exit  
 F8A4' 30 0759 1392 40\$: BSBW PUTERROR ; Give error message  
 04 075C 1393 RET ; Exit  
 0750 1394 :  
 0750 1395 LOADCODE:  
 00002444'EF DD 0750 1396 PUSHL BOOSGL\_CONDRV ; File name string  
 6E D6 0763 1397 INCL (SP) ; Get past the count  
 7E 00002444'FF 9A 0765 1398 MOVZBL @BOOSGL\_CONDRV,-(SP) ; Length of file name  
 57 SE D0 076C 1399 MOVL SP,R7 ; Address of descriptor for file name  
 F88E' 30 076F 1400 BSBW BOOSUFOOPEN ; Open the file for user access (default SYS  
 22 50 E9 0772 1401 BLBC R0,10\$ ; Error  
 SE 08 C0 0775 1402 ADDL #8,SP ; Clean up stack

00002200'EF	9F	0778	1403	PUSHAB	BOOSAB_LOADBUF	: Use code buffer for return address array		
51	DD	077E	1404	PUSHL	R1	: Channel		
50	02	DD	0780	PUSHL	#2	: Arg count		
50	SE	DD	0782	MOVL	SP, R0			
			0785	SCMKRNL_S	ROUTIN = EXESLOAD_CODE,-			
			0785		ARGLIST = (R0)			
04	50	E8	0794	1409	BLBS	R0, 20\$		
F866	30	0797	1410	10\$: BSBW	PUTERROR			
	04	079A	1411	RET				
		079B	1412					
EA	50	E9	07AA	1413	20\$: SCMKRNL_S	ROUTIN = LINK_CODE		
		07AD	1414		BLBC	R0, 10\$		
		07AE	1415		RET			
		07AE	1416					
		001C	07AE	1417	LINK_CODE:			
52	00002200'GF	00	0780	1419	.WORD	^H<R2,R3,R4>		
54	52	00	0787	1420	MOVL	G^BOOSAB_LOADBUF, R2	: Address of loaded code	
53	10	A4	00	078A	1421	MOVL	R2, R4	: Save address of loaded code
00000000'GF	16	07BE	1422	MOVL	SLVSA SYSVECS(R4), R3	: Get address of vectors in SYS.EXE		
10	50	E9	07C4	1423	JSB	G^EXESLINK_VEC	: Connect vectors to loaded routines.	
50	04	A4	DE	07C7	1424	BLBC	R0, 10\$	: Leave on error
03	13	07D2	1425	MOVAL	G^BOOSGL_LOAD_ARGS, AP	: Argument list for initialization routine		
6044	16	07D4	1427	MOVL	SLVSL_INITRTN(R4), R0	: Possible initialization routine		
	04	07D7	1428	10\$: BEQL	10\$	: None, leave		
		07D8	1429	JSB	(R0)[R4]	: Call it		

07D8 1431 : .SBttl BOOSGIVEHELP - Print Help information  
07D8 1432 : Print Help Information  
07D8 1433 :  
07D8 1434 :  
003C 07D8 1435 .Entry BOOSGIVEHELP, ^M<R2,R3,R4,R5> :  
07DA 1436 :  
00000000'GF 9F 07D/ 1437 PUSHAB G^LIB\$GET\_INPUT : Input routine  
00002574'EF 9F 07E0 1438 PUSHAB HELP FLAG : Flags  
00002559'EF 9F 07E6 1439 PUSHAB L^HECP FILE : Library  
08 AC B0 07EC 1440 MOVL TPASL STRINGCNT(AP),- : Set length  
00002578'EF 07EF 1441 HELP DESC  
0C AC D0 07F4 1442 MOVL TPASC STRINGPTR(AP),- : Set address  
0000257C'EF 07F7 1443 HELP DESC+4 : Input string  
00002578'EF 9F 07FC 1444 PUSHAB HELP DESC : Width  
7E D4 0802 1445 CLRL -(SPT) : Output routine  
00000000'GF 9F 0804 1446 PUSHAB G^LIB\$PUT\_OUTPUT : Call help routine  
00000000'GF 06 FB 080A 1447 CALLS #6,G^LBR\$OUTPUT\_HELP  
0811 1448 :  
04 0811 1449 RET : Return with status  
0812 1450 :  
0812 1451 .END :

SST1	= 00000001		BOOSCONUNITS	00000443 RG 05
SST2	= 00000005		BOOSCONVECOFFSET	00000334 RG 05
SCLI.	= 00002464 R 04		BOOSDEVNAME	000003A4 RG 05
SCLI..	= 00002480 R 04		BOOSEXEOPEN	***** X 05
ACFSB_AFLAG	= 0000000B		BOOSFILECLOSE	***** X 05
ACFSB_AUNIT	= 0000000A		BOOSFILEOPEN	***** X 05
ACFSB_CNUMVEC	= 0000001E		BOOSGBFILELEN	000024FD RG 04
ACFSB_COMBO_CSR_OFFSET	= 00000020		BOOSGIBHELP	000007D8 RG 05
ACFSB_COMBO_VECTOR_OFFSET	= 0000001F		BOOSGL_CMDOPT	***** X 06
ACFSB_NUMUNIT	= 00000021		BOOSGL_COMBO_CSR_OFFSET	00002420 RG 04
ACFSCL_LENGTH	= 00000028		BOOSGL_COMBO_VECTOR_OFFSET	0000241C RG 04
ACFSGL_CRB	= 0000240C RG 04		BOOSGL_CONADP	00002424 RG 04
ACFSGL_DDB	= 00002400 RG 04		BOOSGL_CONAUNIT	0000243C RG 04
ACFSGL_DPT	= 00002414 RG 04		BOOSGL_CONCRB	00002454 RG 04
ACFSGL_IDB	= 00002408 RG 04		BOOSGL_CONCREG	00002428 RG 04
ACFSGL_LASTDDB	= 00002410 RG 04		BOOSGL_CONCUNIT	0000242C RG 04
ACFSGL_SB	= 00002418 RG 04		BOOSGL_CONDEV	00002440 RG 04
ACFSGL_UCB	= 00002404 RG 04		BOOSGL_CONDRV	00002444 RG 04
ACFSL_ADAPTER	= 00000000		BOOSGL_CONFLAGS	00002458 RG 04
ACFSL_CONF1GREG	= 00000004		BOOSGL_CONNUMU	00002430 RG 04
ACFSL CONTRLREG	= 0000000C		BOOSGL_CONNUMV	00002438 RG 04
ACFSL_DEVNAME	= 00000014		BOOSGL_CONUNITS	00002448 RG 04
ACFSL_DRVNAME	= 00000018		BOOSGL_CONVECT	00002434 RG 04
ACFSM_RELOAD	= 00000001		BOOSGL_DOT	***** X 05
ACFSV_CRBBLT	= 00000001		BOOSGL_FILEADDR	000024F9 RG 04
ACFSV_GETDONE	= 00000005		BOOSGL_LOAD_ARGS	000025BC RG 04
ACFSV_NOLOAD_DB	= 00000003		BOOSGL_NEXTSTR	0000245C RG 04
ACFSW_AVECTOR	= 00000008		BOOSGL_PARINUSE	000024FE RG 04
ACFSW_CUNIT	= 00000012		BOOSGL_RETSAVE	000024C4 RG 04
ACFSW_CVECTOR	= 00000010		BOOSGL_SELECT	00002460 RG 04
ACFSW_MAXUNITS	= 0000001C		BOOSGQ_CMDESC	0000246C RG 04
ADPSL_AVECTOR	= 0000001C		BOOSGQ_CONSID	0000244C RG 04
ADPSL_CSR	= 00000000		BOOSGQ_LIMITS	000024B4 RG 04
ADPSL_LINK	= 00000004		BOOSGQ_RETADR	000024BC RG 04
ADPSW_TR	= 0000000C		BOOSGT_ACTIVE	0000250A RG 04
AUTOLOG	0000028F RG 06		BOOSGT_CURRENT	00002502 RG 04
BOOSAB_LOADBUF	00002200 R 04		BOOSGT_CVNAME	000024DE RG 04
BOOSAB_PATCH	00000000 RG 04		BOOSGT_DDDNAME	000024F0 RG 04
BOOSAB_PRMBUF	00000200 RG 04		BOOSGT_DEFAULT	00002511 RG 04
BOOSAL_ACF	0000248C RG 04		BOOSGT_DXNAME	000024E7 RG 04
BOOSAL_CLIBLK	00002464 RG 04		BOOSGT_FILE	00002519 RG 04
BOOSA_PRMBLK	***** X 06		BOOSGT_OPNAME	000024DA RG 04
BOOSCONADP	0000031F RG 05		BOOSGT_PROMPT	00002480 RG 04
BOOSCONAUNIT	00000372 RG 05		BOOSGT_SAVE_DEVNAME	00002614 R 04
BOOSCONCNUM	00000367 RG 05		BOOSGT_SYSNAME	***** X 05
BOOSCONCREG	0000034A RG 05		BOOSHICIM	00000000 RG 03
BOOSCONCSROFFSET	0000033F RG 05		BOOSLOAD	000006D8 RG 05
BOOSCONCVEC	00000357 RG 05		BOOSLOCK_GEN	***** X 06
BOOSCONDVRNAME	00000370 RG 05		BOOSLOLIM	00000000 RG 02
BOOSCONF1GALL	00000089 RG 06		BOOSMAKLIST	000002EE RG 05
BOOSCONF1GONE	000000F8 RG 06		BOOSMSCP_ARG	000002DB RG 05
BOOSCONNECT	000004AB RG 05		BOOSMSCP_RESET	0000029A RG 05
BOOSCONNADP	0000032A RG 05		BOOSREADFILE	***** X 05
BOOSCONRESET	00000200 RG 05		BOOSRELOAD	000006DE RG 05
BOOSCONSOLE	00000464 RG 05		BOOSRESETLIST	00000262 RG 05
BOOSCONSYSID_HIGH	00000459 RG 05		BOOSSEARCH	***** X 05
BOOSCONSYSID_LOW	0000044E RG 05		BOOSSENDOPER	00000140 RG 05

- SYSGEN UTILITY AND PARAMETER FILE EDIT				16-SEP-1984 00:13:54		VAX/VMS Macro V04-00	
				14-SEP-1984 16:09:11		[BOOTS.SRC]SYSGEN.MAR;3	
BOOSSETASCII	*****	X	05	IOGEN\$LOADER	*****	X	06
BOOSSETVALUE	*****	X	05	IPLS\$CHED	= 00000003		
BOOSUFOOPEN	*****	X	05	JPIS\$PID	= 00000319		
BOOSUNLOCK_GEN	*****	X	06	LBR\$OUTPUT_HELP	*****	X	05
BOOSUSEACT	000000E9	RG	05	LF	= 0000000A		
BOOSUSEFILE	00000000	RG	05	LIB\$GET_INPUT	*****	X	05
BOOSWRTACT	00000000	RG	06	LIB\$PUT_OUTPUT	*****	X	05
BOOSWRTCUR	00000108	RG	05	LINK_CODE	000007AE	R	05
BOOSWRTSYSFILE	*****	X	05	LOAD\$CODE	0000075D	R	05
BOOCMD\$V_AUTOLOG	= 0000000C			LOAD\$DRV	000006E3	R	05
BOOCMD\$V_EXCLUDE	= 00000007			LOC\$ADP	00000163	R	06
BOOCMD\$V_SELECT	= 00000006			MMG\$A_SYS\$PARAM	*****	X	05
CLISB_RQTYPE	= 00000000			MSCP\$ARG_LIST	0000258C	R	04
CLISC_REQDESC	= 00000001C			MSCP\$ARG_LIST_SIZE	= 00000030		
CLISK_GETCMD	= 00000001			MSCP\$NAME	000025EC	R	04
CLISW_RQSIZE	= 00000008			NAM\$B_RSL	*****	X	05
CONFIGADP	0000018C	RG	06	NAM\$L_RSA	*****	X	05
CONFIG_EXIT	000000E6	R	06	NEXT\$ADP	0000013B	R	06
CONNECT	000005AB	RG	05	OPCSL_MS_TEXT	= 00000008		
CONN\$ADP	00000579	RG	05	OPCSM_NM_CENTRL	= 00000001		
CONN\$ADAP	00000536	RG	05	OPCS_RQ_RQST	= 00000003		
CONSNAME	000024D6	R	04	OPER\$GET\$JPI	00002694	R	04
CR	= 0000000D			OPER\$MSG	000026C0	R	04
CRBSL_INTD	= 00000024			OPER\$MSG\$BUF	000026C8	R	04
CTL\$GL\$PCB	*****	X	06	OPER\$MSG\$FAO	000026AC	R	04
CTR\$TR_AUTOLOG	000025F1	R	04	OPER\$MSG\$ID	000026A8	R	04
CTR\$TR_AUTOLOG_UNIT	00002600	R	04	OPER\$MSG\$NAM	000026B4	R	04
DBBSL_OCB	= 00000004			OPER\$MSG\$PID	000026B0	R	04
DBBST_DRVNAME	= 00000024			OPER\$MSG\$TXT	000026D0	R	04
DONE	000000A9	R	05	OPER\$MSG\$VEC	000026A4	R	04
DYN\$C_LOADCODE	= 00000062			OPER\$NAME\$DESC	000026B8	R	04
EXESA_SYS\$PARAM	*****	X	05	OUT\$BUF	00002628	R	04
EXESC_SYS\$PARSZ	*****	X	05	OUT\$BUF_STR	00002630	R	04
EXESGL_DEF\$FLAGS	*****	X	06	OUT\$LEN	00002610	R	04
EXESGL_DYNAMIC_FLAGS	*****	X	05	OUT\$LEN_UNIT	0000260C	R	04
EXESGL_FLAGS	*****	X	06	PRS\$IPC	= 00000012		
EXESGL_SCB	*****	X	05	PRMS\$B_POS	= 00000015		
EXESGT_STARTUP	*****	X	05	PRMS\$B_SIZE	= 00000014		
EXESLINK_VEC	*****	X	05	PRMS\$C_LENGTH	= 00000032		
EXESLOAD_CODE	*****	X	05	PRMS\$L_ADDR	= 00000000		
EXESV_NOAUTOCNF	*****	X	06	PRMS\$L_FLAGS	= 00000010		
EXESV_WRITE\$SYS\$PARAMS	*****	X	05	PRMS\$M_DYN\$FLAGS	*****	X	06
FACNAME	000024D0	R	04	PRMS\$V_ASCII	= 00000010		
FACNAME\$D	000024C8	RG	04	PRMS\$V_DYNAMIC	= 00000000		
FACNAME\$SZ	= 00000006			PUT\$ERROR	*****	X	06
FF	= 0000000C			RIO\$AB_BUFFER	*****	X	06
FULL_NAME_PTR	00002588	RG	04	RIO\$GW_OUT\$LEN	*****	X	06
HELP_DESC	00002578	R	04	RIO\$OUTPUT_LINE	*****	X	06
HELP_FILE	00002559	R	04	RIO_INP\$NAM	*****	X	05
HELP_FLAG	00002574	R	04	SAVE\$DOT	00002584	R	04
HLPS\$PROMPT	= 00000001			SB\$B_SYSTEM\$ID	= 00000018		
IDBSL\$ADP	= 00000014			SCH\$TIOLOCKR	*****	X	06
IOC\$SAUTOCONFIG	*****	X	06	SCH\$TIOUNLOCK	*****	X	06
IOC\$AUTORESET	*****	X	06	SCSS\$GA_LOCAL\$SB	*****	X	06
IOC\$GL\$ADPLIST	*****	X	06	SCSS\$GL_CDL	*****	X	05
IOC\$SEARCH\$ALL	*****	X	06	SCSS\$GL_MSCP	*****	X	05
IOGEN\$CONSOLE	*****	X	05	SELECT	0000025A	R	06

SGN\$GET_DEVICE	00000348	RG	06
SGN\$GET_DEVICE_LOCK_IODDB	0000035D	R	06
SLVSA_S\$VSCS	= 00000010		
SLVSB_TYPE	= 0000000A		
SLVSL_INITRTN	= 00000004		
SSS_DEVACTIVE	= 000002C4		
SSS_DEVOFFLINE	= 00000084		
SSS_NOPRIV	= 00000024		
SSS_NORMAL	= 00000001		
SSS_NOSUCHDEV	= 00000908		
SYSSCMEEXEC	*****	GX	06
SYSSCMKRL	*****	GX	06
SYSSFAD	*****	X	06
SYSSGETJPI	*****	GX	05
SYSSPUTMSG	*****	GX	05
SYSS SNDOPR	*****	GX	05
SYSGS_CONFQUAL	= 007C808A		
SYSGS_INVADAP	= 007C80BA		
SYSGS_NOADAPTER	= 007C80D2		
SYSGS_NOAUTOCNF	= 007C8002		
SYSGS_NODEV	= 007C9010		
SYSGS_NOTPARAM	= 007C80EA		
SYSGS_WRITEACT	= 007CA013		
SYSGS_WRITECUR	= 007CA01B		
TPASL_NUMBER	= 0000001C		
TPASL_PARAM	= 00000020		
TPASL_STRINGCNT	= 00000008		
TPASL_STRINGPTR	= 0000000C		
TPASL_TOKENCNT	= 00000010		
TPASL_TOKENPTR	= 00000014		
UBA_I0BASE	= 00001000		
UCBSL_CRB	= 00000024		
UCBSL_LINK	= 00000030		
UCBSW_UNIT	= 00000054		
VALID_PAR_FILE	00002580	R	04
VECSL_IDB	= 00000008		

-----  
! Psect synopsis !  
-----

Psect name	Allocation	Psect No.	Attributes
ABS .	00000000	( 0.) 00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000	( 0.) 01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SSSS000	00000000	( 0.) 02 ( 2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC BYTE
ZZZ	00000000	( 0.) 03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE
NOPAGED_DATA	000027D0	(10192.) 04 ( 4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
PAGED_CODE	00000812	( 2066.) 05 ( 5.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
NONPAGED_CODE	00000423	( 1059.) 06 ( 6.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.09	00:00:00.49
Command processing	138	00:00:00.74	00:00:03.19
Pass 1	573	00:00:24.19	00:00:47.20
Symbol table sort	0	00:00:03.72	00:00:06.05
Pass 2	296	00:00:05.60	00:00:13.47
Symbol table output	31	00:00:00.26	00:00:00.27
Psect synopsis output	4	00:00:00.04	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1081	00:00:34.65	00:01:10.73

The working set limit was 2000 pages.

137940 bytes (270 pages) of virtual memory were used to buffer the intermediate code.

There were 130 pages of symbol table space allocated to hold 2287 non-local and 103 local symbols.

1451 source lines were read in Pass 1, producing 130 object records in Pass 2.

50 pages of virtual memory were used to define 47 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
\$255SDUA28:[BOOTS.OBJ]BOOTS.MLB;1	1
\$255SDUA28:[SYS.OBJ]LIB.MLB;1	18
\$255SDUA28:[SYSLIB]STARLET.MLB;2	25
TOTALS (all libraries)	44

2422 GETS were required to define 44 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:SYSGEN/OBJ=OBJ\$:SYSGEN MSRC\$:SYSGEN/UPDATE=(ENH\$:SYSGEN)+EXECMLS/LIB+LIB\$:BOOTS.MLB/LIB

0041 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

